Nos. 2023-1850, -2038

United States Court of Appeals
for the
Fourth Circuit

HONEYWELL INTERNATIONAL INC.; HAND HELD PRODUCTS, INC.; METROLOGIC INSTRUMENTS, INC.,

Plaintiffs-Appellants/Cross-Appellees,

v.

OPTO ELECTRONICS CO., LTD.,

Defendant-Appellee/Cross-Appellant.

On Appeal from the United States District Court for the Western District of North Carolina Case No. 3:21-cv-506-KDB-DCK

JOINT APPENDIX
VOLUME 6 OF 16 (PAGES 2546-3030)

Kirk T. Bradley
M. Scott Stevens
S. Benjamin Pleune
Stephen R. Lareau
Nicholas C. Marais
Lauren N. Griffin
Brandon C.E. Springer
ALSTON & BIRD LLP
Vantage South End
1120 South Tryon Street, Ste. 300
Charlotte, NC 28203
(704) 444-1000

Attorneys for Plaintiffs-Appellants Honeywell International Inc.; Hand-Held Products, Inc.; and Metrologic Instruments, Inc.

York M. Faulkner
YORKMOODYFAULKNER
711
Tensho Building, 7F
3-17-11 Shibaura
Tokyo, Minato-Ku, 108-0023
Japan
(202) 251-0837

Brian D. Schmalzbach
MCGUIREWOODS, LLP
800 East Canal Street
Richmond, Virginia 23219
(804) 775-4746

Attorneys for Defendant-Appellee Opto Electronics Co., Ltd.

April 1, 2024

VOLUME 1 OF 16		
Date	**Description**	**Page**
	Docket Sheet	JA1
2021.09.24	Complaint for Breach of Contract	JA51
2022.04.14	Order on Motion to Dismiss	JA102
2022.08.29	Defendant's Objections and Responses to Plaintiffs' First Set of Requests for Admission	JA108
2023.02.16	Order on Discovery Disputes	JA121
2023.02.22	Honeywell's Motion for Partial Summary Judgment and Legal Determinations Regarding Contractual Interpretation	JA125
2023.02.22	EXHIBIT C – MDL-2001 Product Specification	JA128
2023.03.08	Defendant OPTO Electronics Co., Ltd.'s Motion for Summary Judgment	JA131
2023.03.08	EXHIBIT 1 – ITC Complaint	JA134
2023.03.08	EXHIBIT 2 – ITC Order	JA189
2023.03.08	EXHIBIT 3 – ITC Order and Honeywell's Response	JA194
2023.03.08	EXHIBIT 4 – Claim Chart Excerpts	JA206
2023.03.08	EXHIBIT 5 – ISO-IEC 15415	JA209
2023.03.08	EXHIBIT 6 – Honeywell Website	JA262
2023.03.08	EXHIBIT 7 – U.S. Patent No. 5,243,655	JA271
2023.03.08	EXHIBIT 8 – ISO-IEC 15438	JA300
2023.03.08	EXHIBIT 9 – Accurate Data	JA417
2023.03.08	EXHIBIT 10 – Honeywell Tech Support Page	JA421
2023.03.15	EXHIBIT D – Wang Expert Report	JA424
2023.03.15	EXHIBIT E – MDI-4050 Product Datasheet	JA437
2023.03.21	Order denying OPTO's Objections to Magistrate Judge's Decision (Dkt. 125)	JA440
2023.03.22	EXHIBIT A – L-22X Marketing Materials	JA452
2023.03.22	EXHIBIT B – U.S. Patent No. 10,140,490	JA455
2023.03.29	Defendant's Reply in Support of Motion for Summary Judgment	JA470
2023.03.28	OPTO Cover Motion for Summary Judgment on Patent Misuse	JA487
2023.03.29	EXHIBIT 2 – U.S. Patent No. 7,159,783	JA490
2023.04.20	Summary Judgment Order	JA512

VOLUME 2 OF 16		
Date	**Description**	**Page**
2023.05.01	Honeywell's Motion for Reconsideration and Offer of Proof of Dkt. 195	JA546
2023.05.01	EXHIBIT S - OPTO Electronics Audit Report	JA549
2023.06.02	Order Denying Honeywell's Motion for Reconsideration	JA565
2023.06.20	Joint Stipulation of Facts	JA575
2023.07.05	OPTO's Motion to Strike Honeywell's Second Amended Objections and Responses to OPTO's First Set of Interrogatories	JA579
2023.07.10	EXHIBIT D – June 5, 2023 E-Mail from J. O'Brien	JA583
2023.07.10	EXHIBIT E – OPTO's Amended Exhibit List	JA585
2023.07.10	EXHIBIT H – Honeywell's First Amended Exhibit List	JA629
2023.07.13	Transcript from July 11, 2023 Pre-Trial Conference	JA643
2023.07.13	Order on Motions in Limine and Pretrial Matters	JA762
2023.07.18	Order Granting OPTO's Motion to Strike Honeywell's Second Amended Objections and Responses to OPTO's First Set of Interrogatories	JA776
2023.07.19	Verdict Form	JA781
2023.07.20	Judgment in Case	JA783
2023.07.25	Stipulation Regarding Jury Verdict Damages	JA784
2023.07.26	Amended Judgment	JA787
2023.08.03	Honeywell's Motion for Fees and Costs	JA788
2023.08.03	Memorandum in support of Honeywell's Motion for Fees and Costs	JA791
2023.08.03	EXHIBIT C – June 26, 2023 E-Mail from R. Muckenfuss	JA807
2023.08.03	EXHIBIT D – OPTO Summary of Financial Results for Fiscal Year End 2022	JA810
2023.08.03	EXHIBIT E – OPTO Financial Statement	JA849
2023.08.03	EXHIBIT F – Honeywell International Inc. et al v. Opticon, Inc. et al, C.A. 1:19-cv-1019-CFC (D. Del.) Complaint	JA870

2023.07.17	Trial Transcript - Volume I	JA903
	Testimony of Taylor Smith:	
	Direct Examination by Mr. Pleune	JA966
	Cross Examination by Mr. Muckenfuss	JA973
	Testimony of Jeremy Christopher Whitley:	
	Direct Examination by Mr. Stevens	JA976
	Cross Examination by Mr. VanHoutan	JA1001
	Redirect Examination by Mr. Stevens	JA1009
	Testimony of Craig Smith:	
	Direct Examination by Mr. Marais	JA1017
	Voir Dire Examination by Ms. Lehman	JA1026
	Direct Examination by Mr. Marais	JA1030
	Cross Examination by Ms. Lehman	JA1055
	Redirect Examination by Mr. Marais	JA1078
VOLUME 3 OF 16		
Date	**Description**	**Page**
2023.07.18	Trial Transcript - Volume II	JA1089
	Testimony of Adam Doane:	
	Direct Examination by Mr. Muckenfuss	JA1102
	Cross Examination by Mr. Stevens	JA1170
	Redirect Examination by Mr. Muckenfuss	JA1193
	Testimony of Paul Chartier:	
	Direct Examination by Mr. VanHoutan	JA1214
	Cross Examination by Mr. Stevens	JA1262
	Redirect Examination by Mr. VanHoutan	JA1292
	Recross Examination by Mr. Stevens	JA1293
	Testimony of Rie Ashihara:	
	Direct Examination by Mr. VanHoutan	JA1297
	Cross Examination by Mr. Stevens	JA1302
	Testimony of Cornelis Stoop:	
	Direct Examination by Mr. Stevens	JA1309
	Cross Examination by Mr. VanHoutan	JA1323
	Redirect Examination by Mr. Stevens	JA1328
	Recross Examination by Mr. VanHoutan	JA1330

	Testimony of Rie Ashihara: Direct Examination by Mr. Stevens Cross Examination by Mr. VanHoutan Redirect Examination by Mr. Stevens	 JA1331 JA1335 JA1336
2023.07.19	Trial Transcript - Volume III Testimony of Yoshiaki Kohmo: Direct Examination by Mr. Stevens Testimony of Jeremy Whitley: Direct Examination by Mr. VanHoutan Cross Examination by Mr. Stevens Redirect Examination by Mr. VanHoutan Testimony of Gregory Adams: Direct Examination by Mr. Houghton Cross Examination by Mr. Stevens	JA1341 JA1346 JA1439 JA1465 JA1487 JA1492 JA1546
VOLUME 4 OF 16		
Date	**Description**	**Page**
2023.07.20	Trial Transcript - Volume IV Testimony of Shigeaki Tanaka: Direct Examination by Mr. Stevens Testimony of David Taylor: Direct Examination by Mr. Springer Voir Dire Examination by Mr. McCamey Direct Examination by Mr. Springer Cross Examination by Mr. McCamey Redirect Examination by Mr. Springer Testimony of Ryan N. Herrington: Direct Examination by Mr. Lareau Cross Examination by Mr. VanHoutan	JA1597 JA1601 JA1611 JA1618 JA1619 JA1636 JA1645 JA1646 JA1660
2023.08.14	Honeywell's Notice of Appeal to the Fourth Circuit	JA1688
2023.08.17	EXHIBIT 3 – Zebra License and Settlement Agreement	JA1691
2023.08.17	OPTO Motion for Judgment as a Matter of Law	JA1712
2023.08.17	Trial Transcript Excerpt Testimony of Jeremy Christopher Whitley: Direct Examination by Mr. Stevens Cross Examination by Mr. VanHoutan	JA1716 JA1720 JA1723

2023.08.17	Trial Transcript Excerpt Testimony of Adam Doane: Direct Examination by Mr. Muckenfuss Cross Examination by Mr. Stevens Testimony of Rie Ashihara: Direct Examination by Mr. VanHoutan Redirect Examination by Mr. Stevens	JA1725 JA1732 JA1735 JA1738 JA1740
2023.08.17	Trial Transcript Excerpt Closing Argument by Mr. Stevens.	JA1742
2023.08.24	Reply in Support of Motion for Fees and Costs	JA1746
2023.08.24	EXHIBIT A – E-mail from Z. McCamey	JA1764
2023.08.31	Honeywell’s Response to OPTO’s Motion for Judgment as a Matter of Law	JA1767
2023.09.27	Order on Post-Trial Motions	JA1797
2023.10.02	Honeywell’s Amended Notice of Appeal to the United States Court of Appeals for the Fourth Circuit	JA1814
2023.10.04	Notice of Appeal of Defendant OPTO Electronics Co., Ltd. to the United States Court of Appeals for the Fourth Circuit	JA1817
2023.10.27	Honeywell’s Notice of Appeal to the United States Court of Appeals for the Federal Circuit	JA1820
2023.11.07	OPTO’s Notice of Appeal to the United States Court of Appeals for the Federal Circuit	JA1823

Exhibit	**Description**	**Page No.**
PX-5	ISO/IEC 15438	JA1827
PX-12	ISO/IEC 15415	JA1943
PX-17	U.S. Patent No. 5,304,786	JA1995
PX-18	MDL-2001 Datasheet	JA2046
PX-19	OPR-2001Z Leaflet	JA2048
PX-62	Auditor’s Report and Internal Control Audit Report	JA2050
PX-63	Notice Regarding Filing of Lawsuit, Recording of Extraordinary Loss, and Revision of Consolidated Earnings Forecast for the Fiscal Year Ending November 2021, OPTO Press Release, November 30, 2021	JA2072
PX-87	NLV-1001 Product Specification	JA2078
PX-125	MDL-1000 Specification	JA2080

PX-188	*In the Matter of Certain Barcode Scanners, Scan Engines, Products Containing the Same, and Components Thereof*, Complaint, Investigation No. 337-TA-1165	JA2082
	VOLUME 5 OF 16	
Exhibit	**Description**	**Page No.**
PX-206	U.S. Patent No. 9,465,970	JA2136
PX-207	U.S. Patent No. 10,140,490	JA2201
PX-212	U.S. Patent No. 7,159,783	JA2215
PX-214	U.S. Patent No. 8,752,766	JA2236
PX-215	U.S. Patent No. 9,230,140	JA2266
PX-216	U.S. Patent No. 10,846,498	JA2280
PX-218	U.S. Patent No. 10,235,547	JA2308
PX-225	U.S. Patent No. 8,587,595	JA2336
PX-226	U.S. Patent No. 9,092,686	JA2348
PX-227	U.S. Patent No. 9,659,203	JA2361
PX-228	U.S. Patent No. 9,384,378	JA2374
PX-319	3Q JASDAQ Regulatory Filing (Sept. 22, 2022)	JA2387
PX-325	OPTO’s Objections and Responses to Honeywell’s First Requests for Admission	JA2410
DX-816	US Patent No 7,387,253	JA2423
DX-818	US Patent No 7,104,456	JA2486
	VOLUME 6 OF 16	
Exhibit	**Description**	**Page No.**
DX-838	US Patent No 7,472,831	JA2546
DX-848	P. Chartier Trial Exhibit 1001	JA2713
DX-849	P. Chartier Trial Exhibit 1002	JA2857
DX-850	P. Chartier Trial Exhibit 1003	JA2987
	VOLUME 7 OF 16	
Exhibit	**Description**	**Page No.**
DX-851	P. Chartier Trial Exhibit 1004	JA3031
DX-852	P. Chartier Trial Exhibit 1005	JA3083
DX-853	P. Chartier Trial Exhibit 1006	JA3199
DX-854	P. Chartier Trial Exhibit 1007	JA3201

VOLUME 8 OF 16 – SEALED		
Date	**Description**	**Page No.**
2022.01.05	Answer, Affirmative Defenses, and Counterclaims	JA3215
2022.04.11	Hearing Transcript on Motion to Dismiss	JA3249
2022.08.11	Defendant's Objections and Responses to Plaintiffs' First Set of Interrogatories	JA3293
2022.09.06	Honeywell's Objections and Responses to Defendant's First Set of Interrogatories	JA3336
2022.09.06	Honeywell's Objections and Responses to Defendant's First Set of Requests for Admission	JA3375
2022.09.21	Deposition Transcript of J. Whitley	JA3393
2022.09.26	Deposition Transcript of R. Ashihara	JA3574
VOLUME 9 OF 16 – SEALED		
Date	**Description**	**Page No.**
2022.09.27	Deposition Transcript of Y. Kohmo	JA3683
2022.09.29	Defendant's Objections and Responses to Plaintiffs' Second Set of Interrogatories	JA3757
2022.09.29	Defendant's Supplemental and Amended Objections and Responses to Plaintiffs' First Set of Interrogatories	JA3769
2022.09.30	Honeywell's First Amended Objections and Responses to Defendant's First Set of Interrogatories	JA3815
2022.10.21	Expert Report of Mr. Ryan N. Herrington	JA3857
2022.11.18	Expert Report of Dr. Greg Adams	JA3923
2023.02.22	Memorandum in Support of Motion for Partial Summary Judgment and Legal Determinations Regarding Contractual Interpretation	JA3950
2023.02.02	EXHIBIT A – License and Settlement Agreement	JA3973
2023.02.22	EXHIBIT B – July 1, 2019 Correspondence	JA4005
2023.02.22	EXHIBIT D – September 21, 2022 J. Whitley Deposition Excerpts	JA4015
2023.02.22	EXHIBIT E – September 29, 2022 OPTO's Supplemental Responses to First Interrogatories	JA4022
2023.02.22	EXHIBIT F – September 29, 2022 OPTO's Objections and Responses to Second Interrogatories	JA4029
2023.02.22	EXHIBIT G - G. Adams Expert Report Excerpts	JA4034

2022.02.25	Transcript of February 15, 2022 Status Hearing and Discovery Conference	JA4043
VOLUME 10 OF 16 – SEALED		
Date	**Description**	**Page No.**
2023.03.02	OPTO's Objections to Discovery Rulings	JA4223
2023.03.08	Defendant's Memorandum in Support of Its Motion for Summary Judgment	JA4250
2023.03.08	EXHIBIT 1 – First Amendment to License and Settlement Agreement	JA4282
2023.03.08	EXHIBIT 2 – Second Amendment to License and Settlement Agreement	JA4285
2023.03.08	EXHIBIT 3 – Application and Declaration for Remittance	JA4291
2023.03.08	EXHIBIT 4 – Application and Declaration for Remittance	JA4293
2023.03.08	EXHIBIT 5 – Quarterly Royalty Reports and Wire Transfer Receipts	JA4295
2023.03.08	Defendant's Response in Opposition to Plaintiffs' Motion for Partial Summary Judgment and Legal Determinations Regarding Contractual Interpretation	JA4333
2023.03.15	Reply in Support of Honeywell's Motion for Partial Summary Judgment and Legal Determinations Regarding Contractual Interpretation	JA4364
2023.03.15	EXHIBIT A – OPTO's Supplemental and Amended Objections and Responses to First Set of Interrogatories	JA4381
2023.03.15	EXHIBIT B – ISO/IEC 15438	JA4398
2023.03.15	EXHIBIT C – September 27, 2022 Excerpts from Deposition of Yoshiaki Kohmo	JA4515
2023.03.16	Honeywell's Response to OPTO's Objections to Magistrate Judge's Decision	JA4523
2023.03.22	Honeywell's Response in Opposition to OPTO's Motion for Summary Judgment	JA4551
2023.03.28	Honeywell's Memo in Support of Motion for Partial Summary Judgment Regarding Patent Misuse	JA4576
2023.03.28	EXHIBIT C – July 1, 2019 Correspondence from B. Pleune to K. Chu	JA4595

2023.03.28	EXHIBIT D – Defendant's Supplemental and Amended Objections and Responses to Plaintiff's First Set of Interrogatories	JA4605
2023.03.29	OPTO's Memorandum in Support of Motion for Summary Judgment Regarding Patent Misuse	JA4616
2023.04.10	Honeywell's Opposition to OPTO's Motion for Summary Judgment Regarding Patent Misuse	JA4632
2023.04.10	EXHIBIT A – September 30, 2022 Honeywell's First Amended Objections & Responses to OPTO's First Set of Interrogatories	JA4655
2023.04.10	OPTO's Response to Honeywell's Patent Misuse Motion for Summary Judgment	JA4674
VOLUME 11 OF 16 – SEALED		
Date	**Description**	**Page No.**
2023.04.10	EXHIBIT A – OPTO's Supplemental Responses to Plaintiffs' First Set of Interrogatories	JA4695
2023.05.01	Memorandum in Support of Honeywell's Motion for Reconsideration of Dkt. 195	JA4742
2023.05.01	EXHIBIT A – License and Settlement Agreement	JA4769
2023.05.01	EXHIBIT B – Honeywell's First Amended Objections and Responses to Defendant's First Set of Interrogatories	JA4801
2023.05.01	EXHIBIT C – Expert Report of Ryan N. Herrington	JA4844
2023.05.01	EXHIBIT D – September 8, 2020 Letter to OPTO Electronics	JA4911
2023.05.01	EXHIBIT E – September 16, 2020 E-mail from Counsel for OPTO	JA4914
2023.05.01	EXHIBIT F – October 23, 2020 E-mail from Counsel for OPTO	JA4917
2023.05.01	EXHIBIT G – December 16, 2020 Letter to Counsel for OPTO	JA4927
2023.05.01	EXHIBIT H – January 1, 2021 E-mail from Counsel for OPTO	JA4929
2023.05.01	EXHIBIT I – Non-Disclosure & Confidentiality Agreement	JA4936
2023.05.12	EXHIBIT J – January 4, 2021 E-mail from Matsuzawa	JA4940

2023.05.12	EXHIBIT K – January 6, 2021 E-mail from Matsuzawa	JA4952
2023.05.12	EXHIBIT L – January 11, 2021 E-mail to Matsuzawa	JA4964
2023.05.12	EXHIBIT M – January 15, 2021 E-mail to Matsuzawa	JA4980
2023.05.12	EXHIBIT N – January 26, 2021 E-mail to Matsuzawa	JA4989
2023.05.12	EXHIBIT O – February 1, 2021 and February 10, 2021 E-mails to Matsuzawa	JA5001
2023.05.12	EXHIBIT P – February 3, 2021 E-mail to Matsuzawa	JA5012
2023.05.12	EXHIBIT Q – March 31, 2021 E-mail to Matsuzawa	JA5018
VOLUME 12 OF 16 – SEALED		
Date	**Description**	**Page No.**
2023.05.01	EXHIBIT R – Deposition transcript of Rie Ashihara	JA5146
2023.05.01	EXHIBIT S – April 13, 2023 Excerpts of Motions	JA5255
2023.05.10	Hearing Transcript from Motions Heard on April 13. 2023	JA5398
2023.05.15	Defendant's Response in Opposition to Plaintiffs' Motion for Reconsideration of Dkt. 195 and Offer of Proof	JA5540
2023.05.22	Honeywell's Reply in Support of Its Motion for Reconsideration and Offer of Proof	JA5568
2023.06.20	OPTO's Proposed Findings of Fact and Conclusions of Law re Patent Misuse	JA5586
2023.06.20	OPTO's Trial Brief	JA5605
VOLUME 13 OF 16 – SEALED		
Date	**Description**	**Page No.**
2023.06.20	Honeywell's Trial Brief Regarding Jury Trial	JA5626
2023.06.20	Honeywell's Trial Brief Regarding Patent Misuse	JA5647
2023.06.20	Honeywell's Proposed Findings of Fact and Conclusions of Law	JA5671
2023.07.05	OPTO's Memorandum in Support of Motion to Strike	JA5690
2023.07.05	EXHIBIT 1 – Honeywell's Second Amended Objections and Responses to Defendant's First Set of Interrogatories	JA5706

2023.07.10	Honeywell's Response to Motion to Strike	JA5753
2023.07.10	EXHIBIT A – Honeywell's First Amended Objections and Responses to Defendant's First Set of Interrogatories	JA5770
2023.07.10	EXHIBIT B - Honeywell's Second Amended Objections and Responses to Defendant's First Set of Interrogatories	JA5813
2023.07.10	EXHIBIT C - Redlined Version of Honeywell's Interrogatory Responses	JA5860
2023.07.10	EXHIBIT F – December 27, 2022 OPTO's Second Supplemental Responses to Honeywell's First Set of Interrogatories	JA5908
2023.07.10	EXHIBIT G – April 12, 2023 Rebuttal Expert Report of Ryan H. Herrington	JA5956
2023.07.10	EXHIBIT I – October 21, 2022 Expert Report of David O. Taylor	JA6014
2023.08.03	EXHIBIT A – License and Settlement Agreement	JA6072
VOLUME 14 OF 16 – SEALED		
Date	**Description**	**Page No.**
2023.08.03	EXHIBIT B – Declaration of M. Scott Stevens	JA6104
2023.08.17	OPTO's Opposition to Honeywell's Motion for Attorney Fees and Costs	JA6263
2023.08.17	EXHIBIT 1 – Fees and Costs Analysis	JA6295
2023.08.17	EXHIBIT 2 – Metrologic Settlement and License Agreement	JA6317
2023.08.17	EXHIBIT 4 – Doc Production Invoice	JA6333
2023.08.17	EXHIBIT 5 – Invoice	JA6336
2023.08.17	OPTO Memo in Support of Motion for Judgment as a Matter of Law	JA6338
2023.08.31	EXHIBIT A – Defendant's Second Supplemental Objections and Responses to Plaintiff's Frist Set of Interrogatories	JA6369
2023.09.12	OPTO Reply in Support of Motion for Judgment as a Matter of Law	JA6417

VOLUME 15 OF 16 – SEALED		
Exhibit	**Description**	**Page No.**
JX-1	License and Settlement Agreement	JA6432
JX-2	First Amendment to License and Settlement Agreement	JA6436
JX-3	Second Amendment to License and Settlement Agreement	JA6465
PX-4	ISO/IEC 15438	JA6470
PX-8	ISO/IEC 24728	JA6586
PX-10	ISO/IEC 24723	JA6710
PX-13	European Pre-Standard, ENV12925	JA6762
VOLUME 16 OF 16 – SEALED		
Exhibit	**Description**	**Page No.**
PX-61	Table of OPTO Interrogatory Response No. 3	JA6878
PX-243	Letter from B. Pleune to Quinn Emanuel (dated July 1, 2019)	JA6925
PX-244	Pleune Correspondence to Goldstein (May 21, 2021)	JA6934
PX-245	Report Draft OP EU 4 April2021.xlsx (Enclosure to Audit Letter)	JA6936
PX-246	Report Draft OP-1 0220.xlsx (Enclosure to Audit Letter)	JA6969
PX-247	Report Draft OP US_20210414.xlsx (Enclosure to Audit Letter)	JA7068
PX-259	E-Mail Chain with R. Goldstein (October 23, 2020)	JA7096
PX-261	E-Mail Chain with R. Goldstein (January 12, 2021)	JA7105
PX-288	Pleune June 2021 correspondence to Goldstein	JA7111
PX-289	Matsuzawa & Co. Invoice in Japanese	JA7113
DX-095	September 16, 2020 Email from Goldstein to Pleune	JA7118
DX-096	September 16, 2020 Letter from Goldstein to Pleune	JA7120
DX-098	Sales Spreadsheets Sent by Goldstein to Pleune	JA7121
DX-111	September 30 – December 16 Email Thread Goldstein and Pleune	JA7175
DX-769	Honeywell's ITC Claims Charts	JA7184

US007472831B2

(12) United States Patent
Schnee

(10) Patent No.: US 7,472,831 B2
(45) Date of Patent: Jan. 6, 2009

(54) **SYSTEM FOR DETECTING IMAGE LIGHT INTENSITY REFLECTED OFF AN OBJECT IN A DIGITAL IMAGING-BASED BAR CODE SYMBOL READING DEVICE**

(75) Inventor: **Michael Schnee**, Aston, PA (US)

(73) Assignee: **Metrologic Instruments, Inc.**, Blackwood, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 57 days.

(21) Appl. No.: **11/210,507**

(22) Filed: **Aug. 23, 2005**

(65) **Prior Publication Data**

US 2006/0011725 A1 Jan. 19, 2006

Related U.S. Application Data

(63) Continuation-in-part of application No. 10/894,478, filed on Jul. 19, 2004, now Pat. No. 7,357,325, which is a continuation of application No. 10/712,787, filed on Nov. 13, 2003, now Pat. No. 7,128,266.

(51) **Int. Cl.**
G06K 7/10 (2006.01)

(52) **U.S. Cl.** **235/454**; 235/462.11; 235/462.22; 235/462.42

(58) **Field of Classification Search** 235/454, 235/462.11, 462.22, 462.25, 462.42
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,812,459	A	5/1974	MacNeill	382/296
4,226,541	A	10/1980	Tisue	365/430
4,335,302	A	6/1982	Robillard	235/462
4,560,862	A	12/1985	Eastman	235/467
4,667,255	A	5/1987	Lindberg	358/293
4,677,302	A	6/1987	Chiu	250/559.22
4,794,239	A	12/1988	Allias	235/462.1
4,796,997	A	1/1989	Svetkoff	356/608
4,850,712	A	7/1989	Abshire	356/602
5,288,985	A	2/1994	Chadima	235/462.45
5,313,070	A	5/1994	Vala	250/559.08

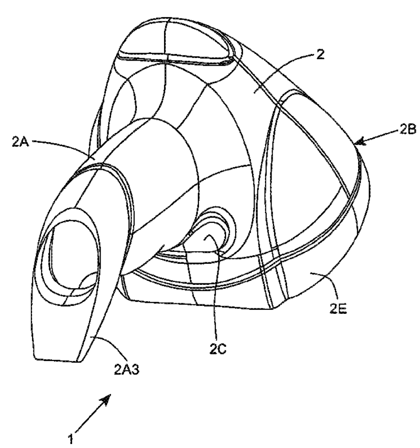
(Continued)

Primary Examiner—Ahshik Kim
(74) *Attorney, Agent, or Firm*—Glenn A. Cavanaugh, Esq.

(57) **ABSTRACT**

A digital imaging-based bar code symbol reading device employing an automatic light exposure measurement and illumination control subsystem having an optical axis which is coincident with the field of view (FOV) of a digital imaging area-type sensing array. The bar code symbol reading device including a system for detecting image light intensity reflected off an object in the FOV of the digital imaging-based bar code symbol reading device, having image formation optics including a beam splitter with a surface of a known reflection/transmission ratio. The image formation optics being arranged such that light reflected off of an object placed in the FOV of the digital imaging-based bar code symbol reading device is directed to the beam splitter whereby a portion of the return light being reflected from the image is directed towards the area-type sensing array during illumination operations in an image capture mode, and a portion of the return light being reflected from the images is transmitted through the beam splitter and focused upon a photodiode for detecting image light intensity and subsequent processing by an automatic light exposure measurement and illumination control subsystem, whereby said automatic light exposure measurement and illumination control subsystem controls illumination intensity produced by said illumination array subsystem. The digital imaging-based bar code symbol reading device further includes a system control subsystem for activating and controlling said subsystem components described above.

7 Claims, 119 Drawing Sheets



U.S. PATENT DOCUMENTS

5,418,357	A		5/1995	Inoue	235/462.11
5,420,635	A		5/1995	Konishi	348/362
5,463,202	A		10/1995	Kurosawa	219/121.83
5,484,994	A		1/1996	Roustaei	235/462.25
5,602,379	A		2/1997	Uchimura	235/462.11
5,623,137	A		4/1997	Powers	235/462.23
5,627,358	A		5/1997	Roustaei	235/462.11
5,744,815	A	*	4/1998	Gurevich et al.	250/566
5,756,981	A		5/1998	Roustaei	235/462.42
5,773,806	A		6/1998	Longacre	235/462.1
5,773,810	A		6/1998	Hussey	235/462.25
5,777,314	A		7/1998	Roustaei	235/462
5,780,834	A		7/1998	Havens	235/462.1
5,783,811	A		7/1998	Feng	235/462.42
5,784,102	A		7/1998	Hussey	348/296
5,786,582	A		7/1998	Roustaei	235/462
5,825,006	A		10/1998	Longacre	235/462.27
5,831,254	A		11/1998	Karpen	235/454
5,912,452	A	*	6/1999	Wiklof et al.	235/472.01
5,929,418	A		7/1999	Ehrhart	235/462
5,932,862	A		8/1999	Hussey	235/462
5,939,709	A		8/1999	Ghislain	250/216
5,942,741	A		8/1999	Longacre	235/462.12
6,038,067	A		3/2000	George	359/368
6,123,261	A		9/2000	Roustaei	235/472.01
6,181,427	B1		1/2001	Yarussi	356/445
6,250,551	B1		6/2001	He	235/462.07
6,290,135	B1	*	9/2001	Acosta et al.	235/472.01
6,347,163	B2	*	2/2002	Roustaei	382/324
6,525,816	B2		2/2003	Asstuen	356/364
6,601,767	B1		8/2003	Gu	235/462.01
6,734,471	B2		5/2004	Kim	257/184
6,863,216	B2		3/2005	Tsikos	235/462.01
2004/0190005	A1		9/2004	Doucet	356/614
2005/0103857	A1		5/2005	Zhu	235/462.22
2005/0103860	A1		5/2005	Zhu	235/462.22
2005/0116040	A1		6/2005	Zhu	235/462.31

* cited by examiner

HONEYWELL-00192846

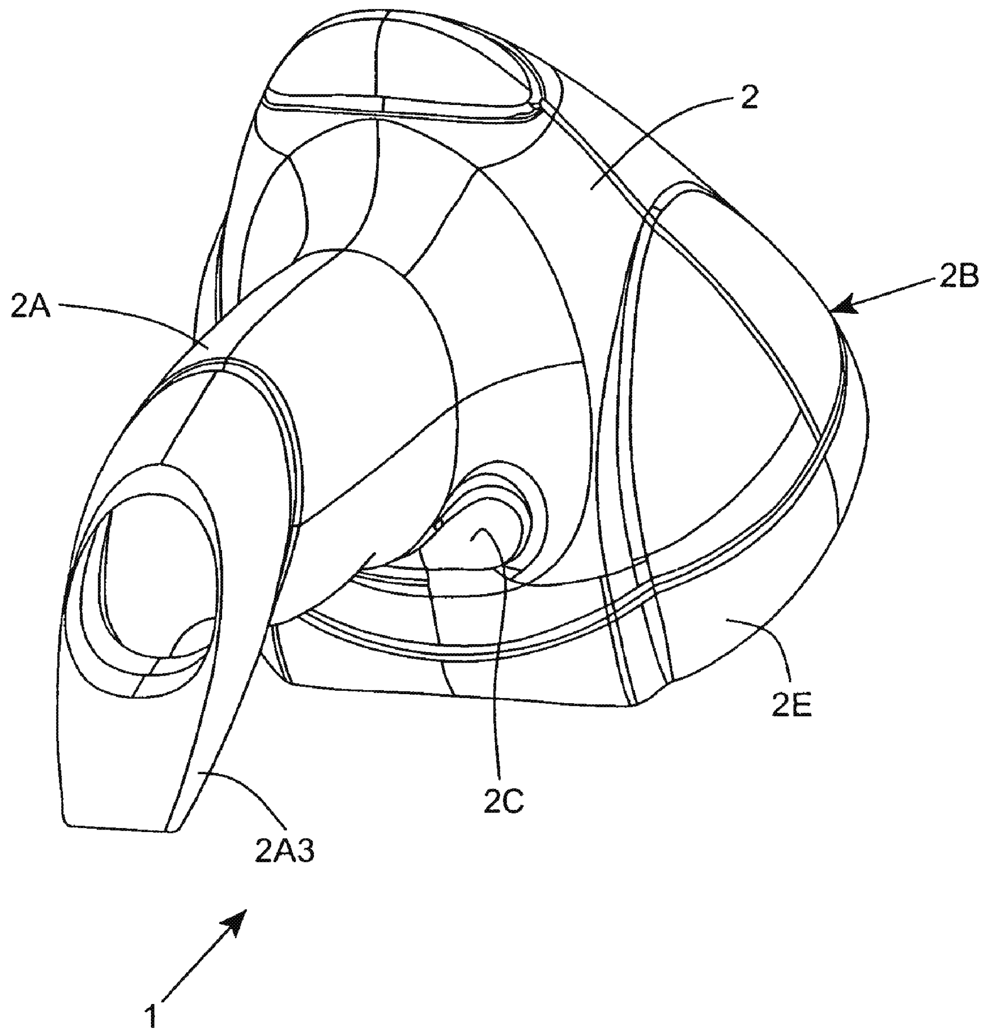


FIG. 1A

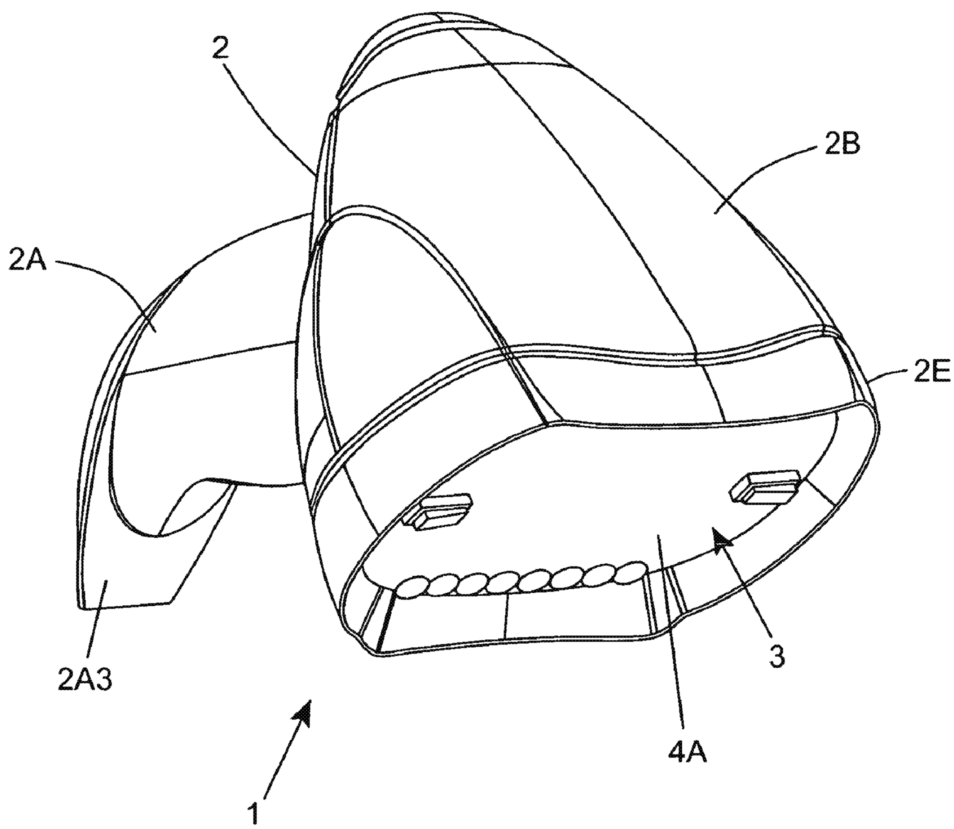


FIG. 1B

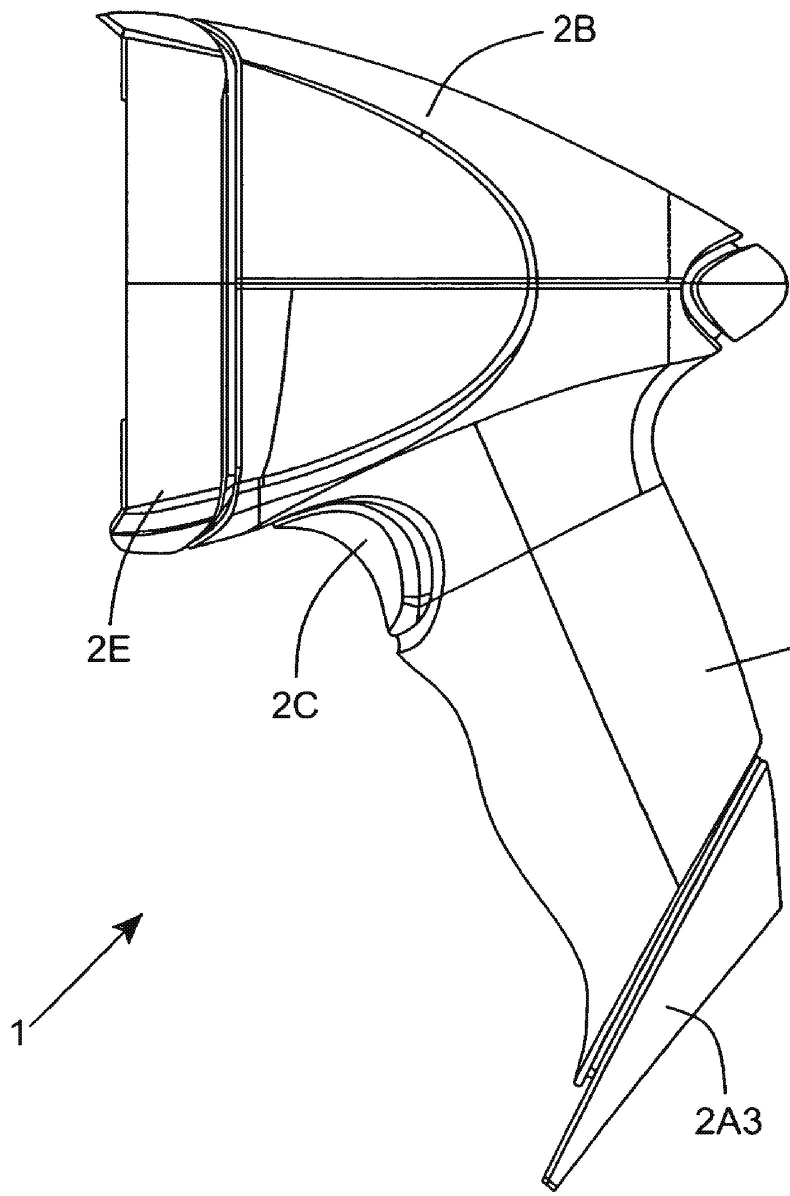


FIG. 1C

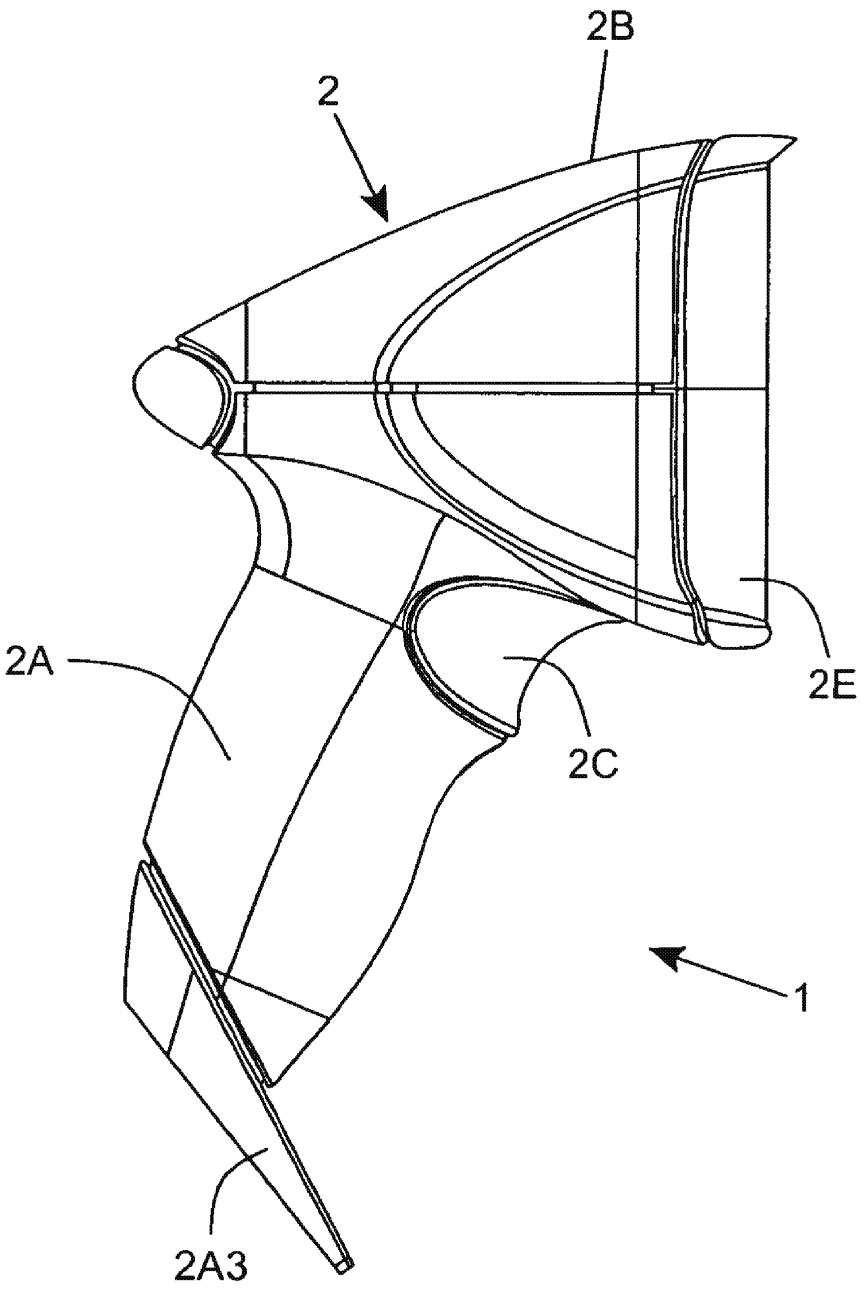


FIG. 1D

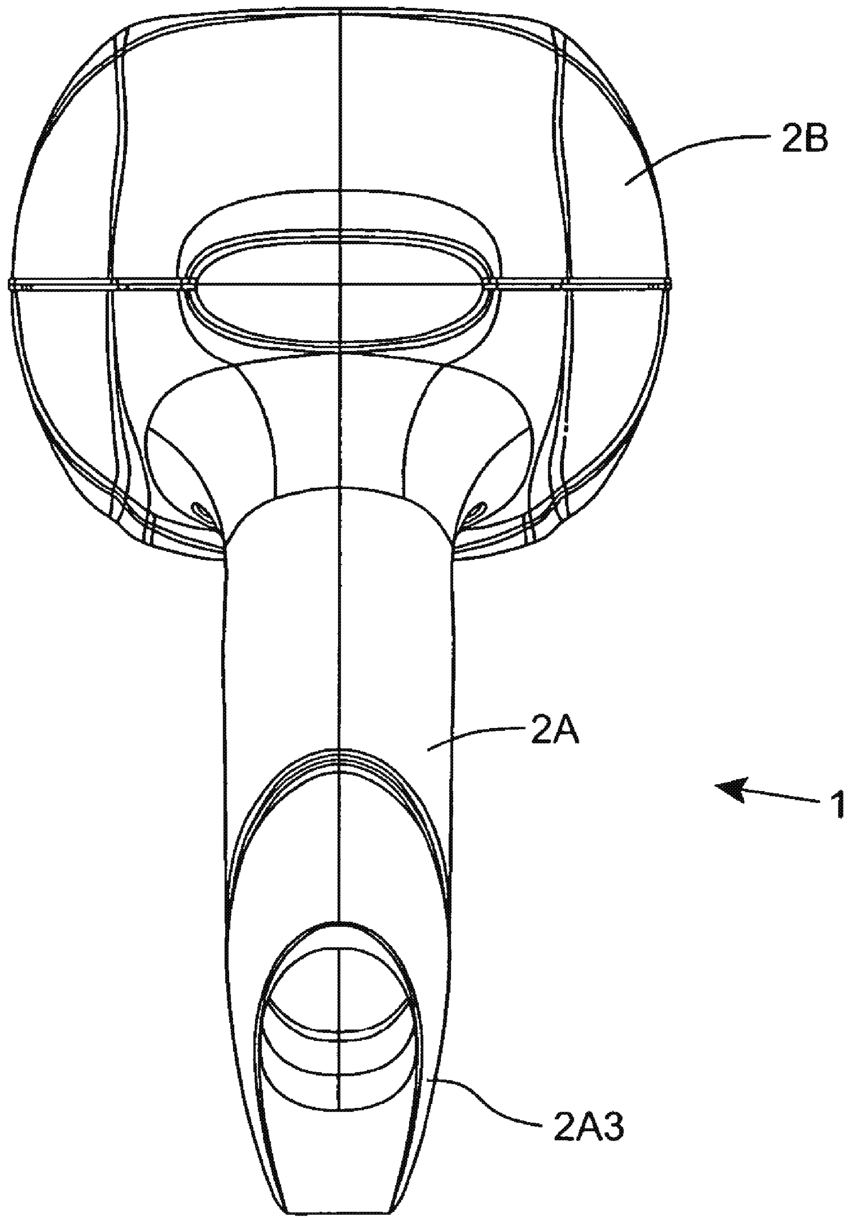


FIG. 1E

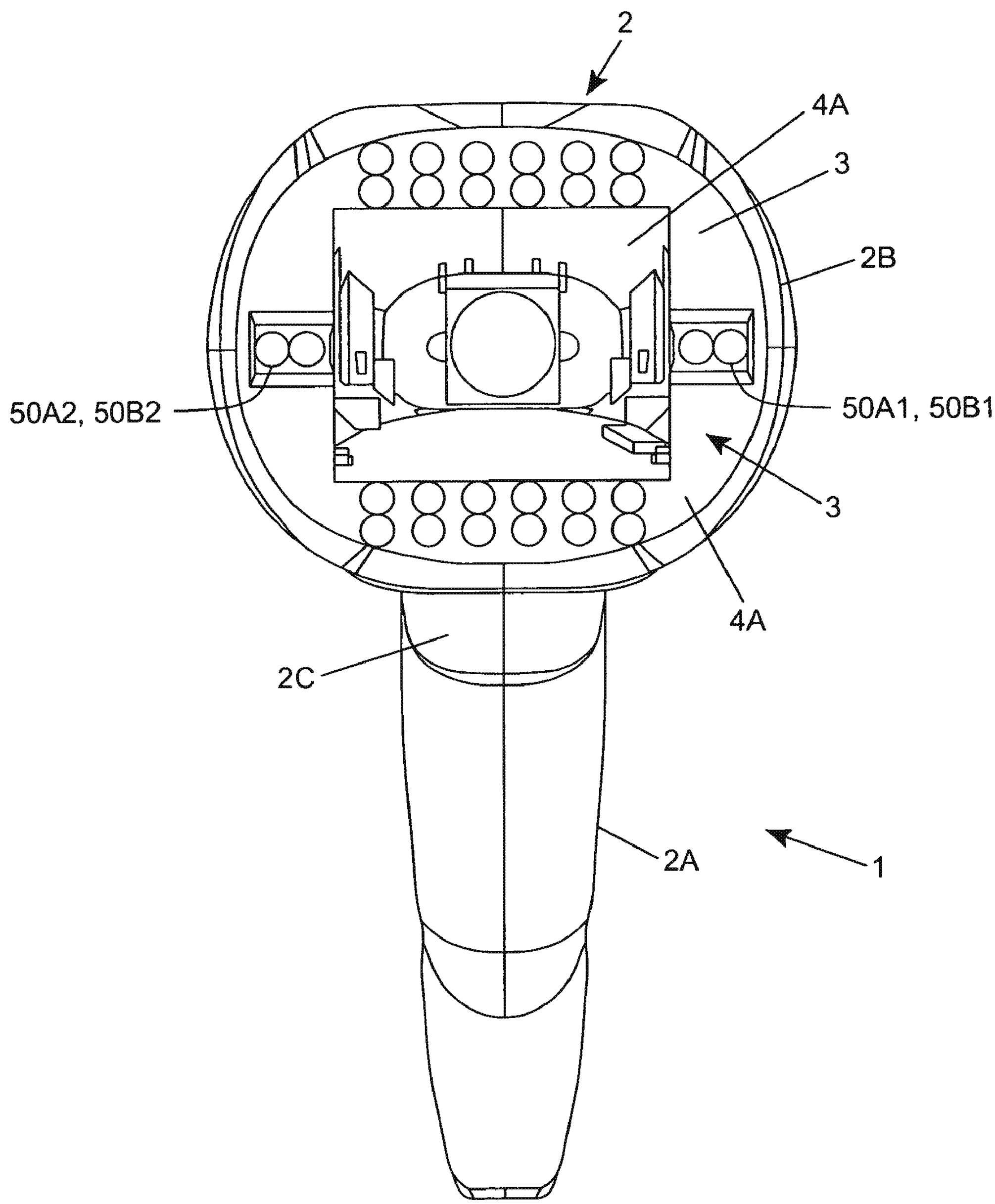


FIG. 1F

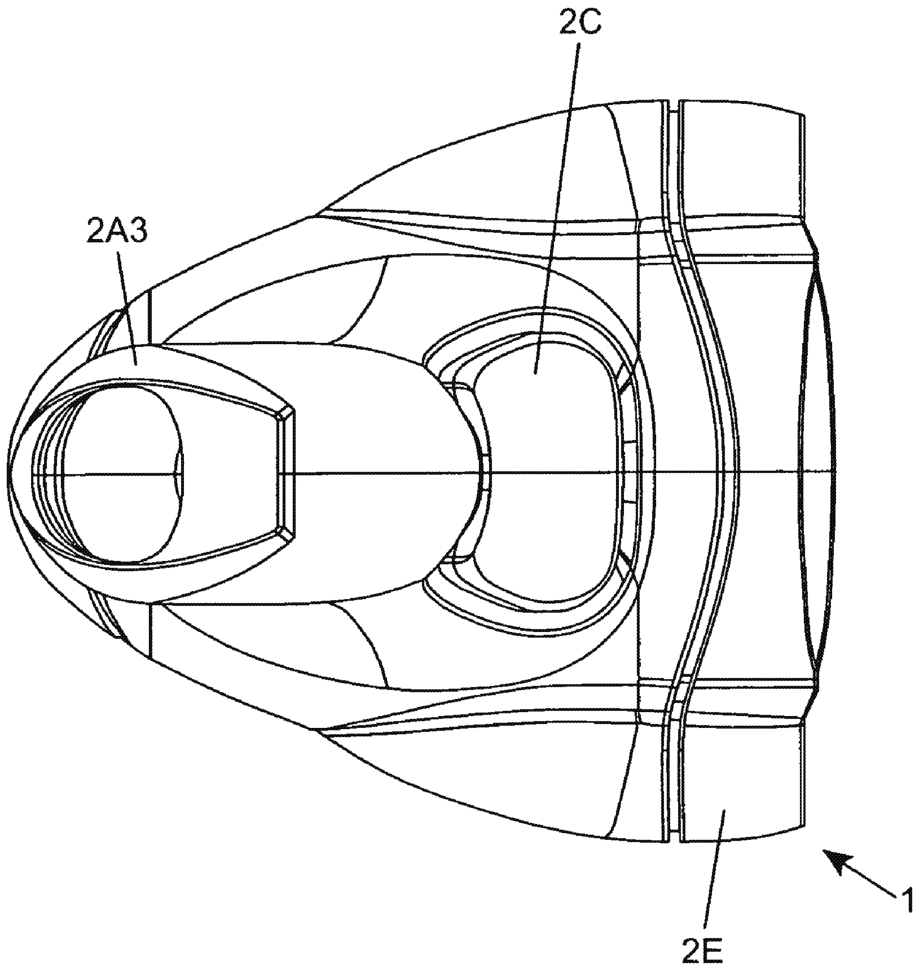


FIG. 1G

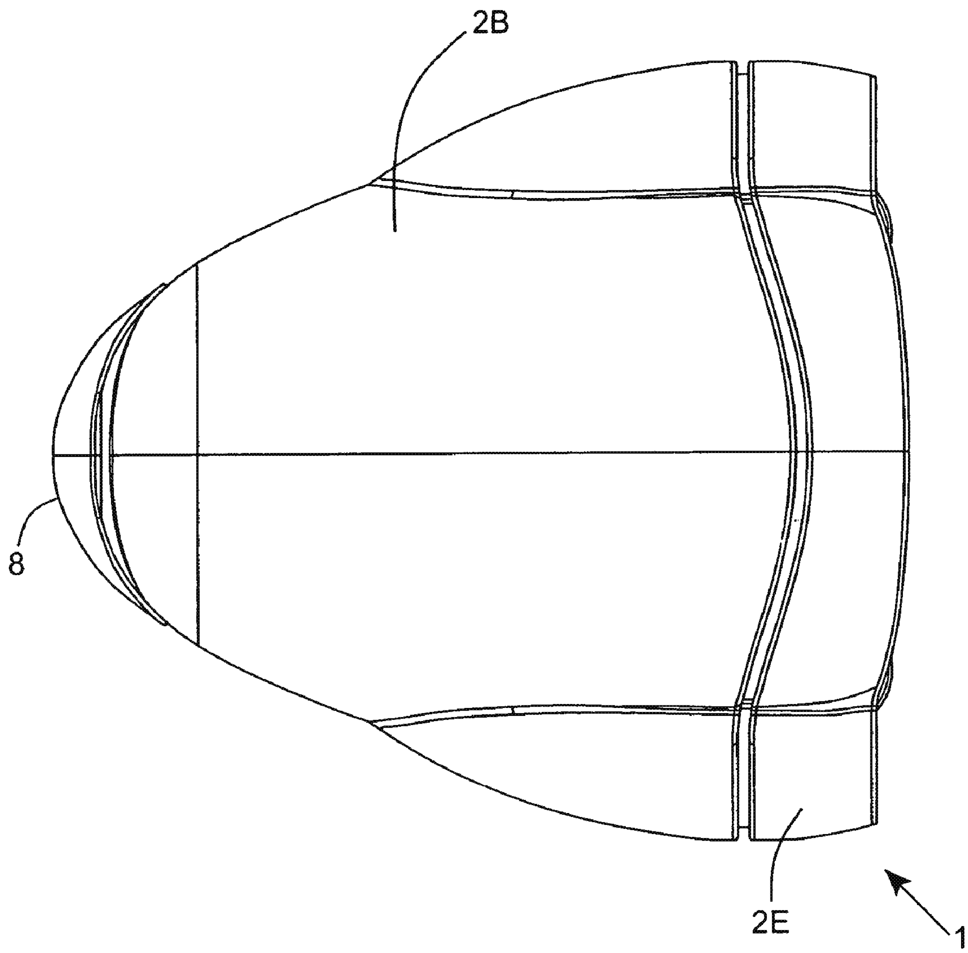


FIG. 1H

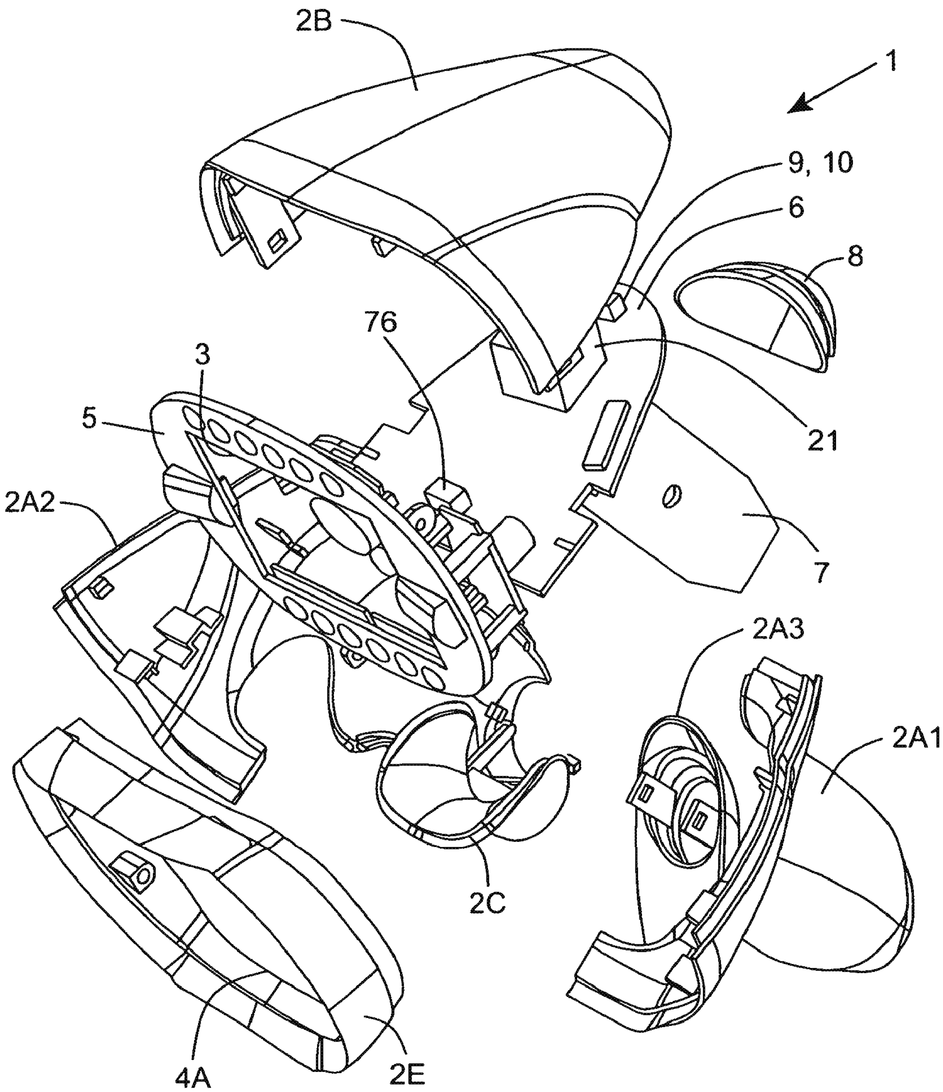


FIG. 1I

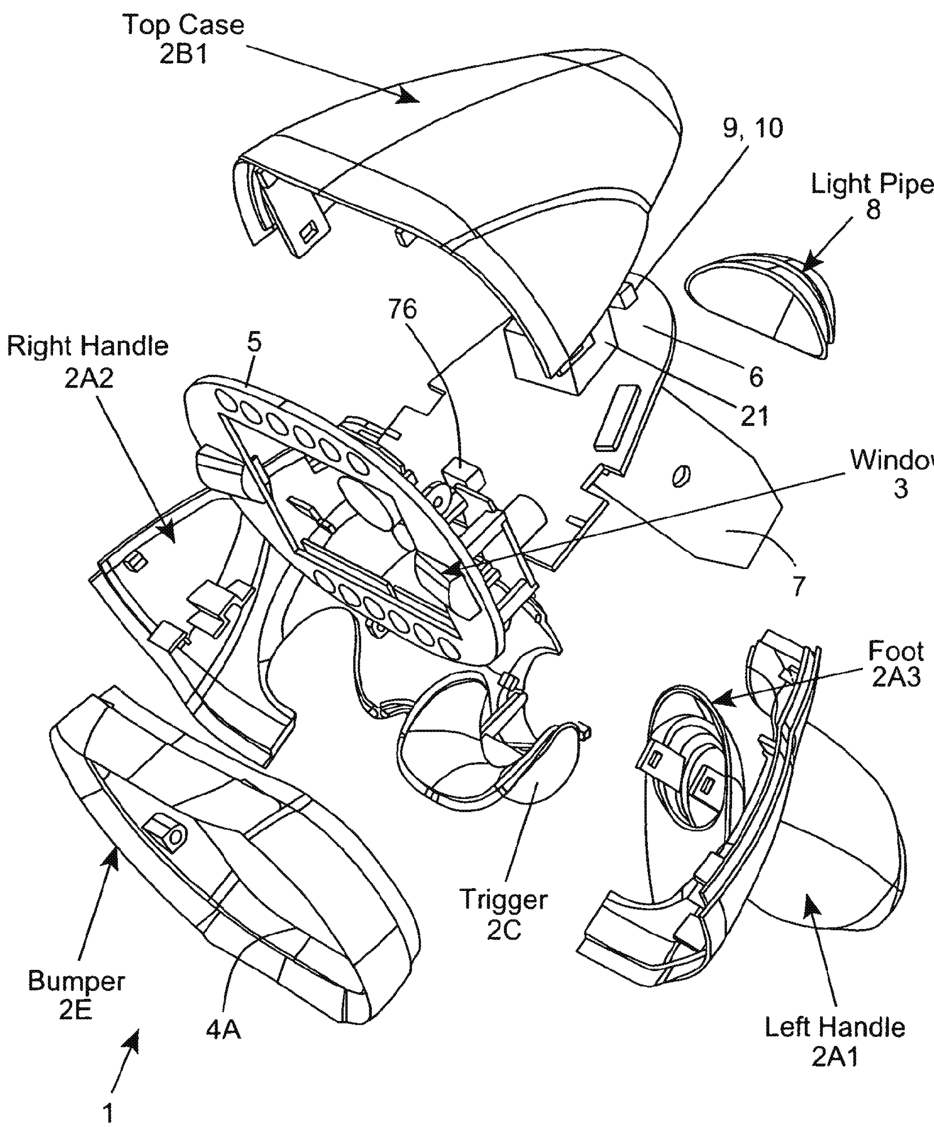

Top Case
2B1
9, 10
Light Pipe
8
76
Right Handle
2A2
5
6
21
Window
3
7
Foot
2A3
Trigger
2C
Bumper
2E
4A
Left Handle
2A1
1

FIG. 1J

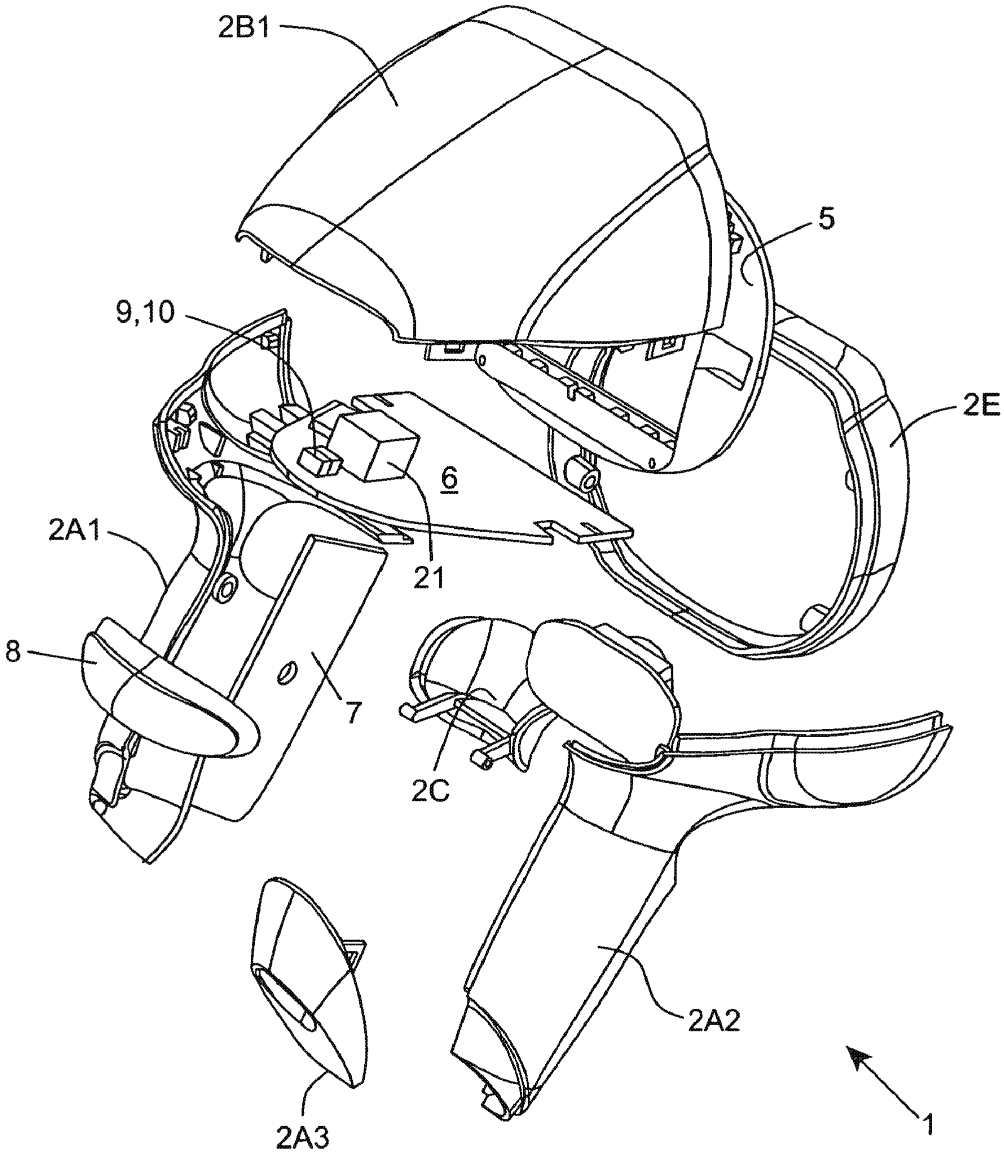


FIG. 1K

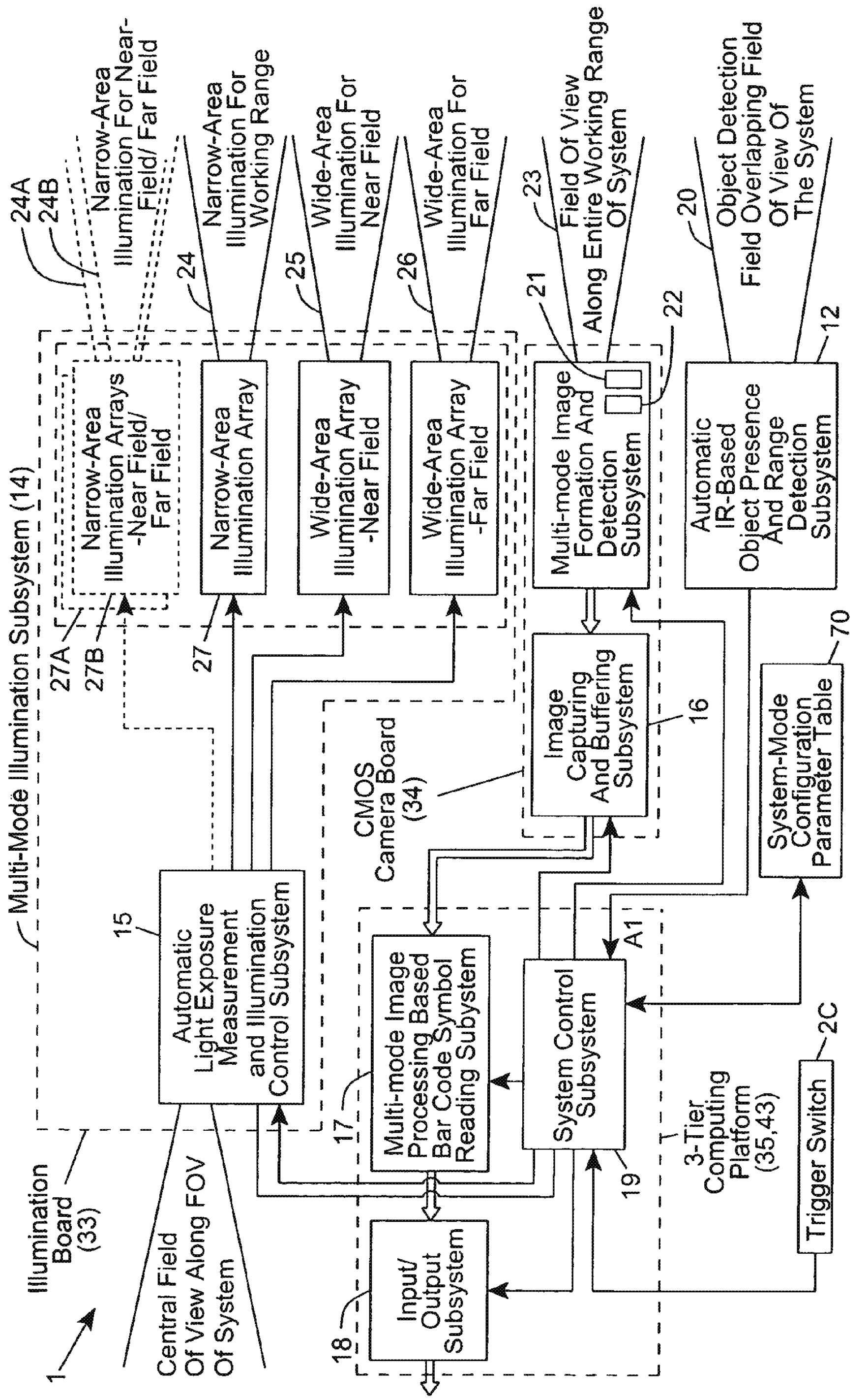


FIG. 2A1

HONEYWELL-00192858

JA2559

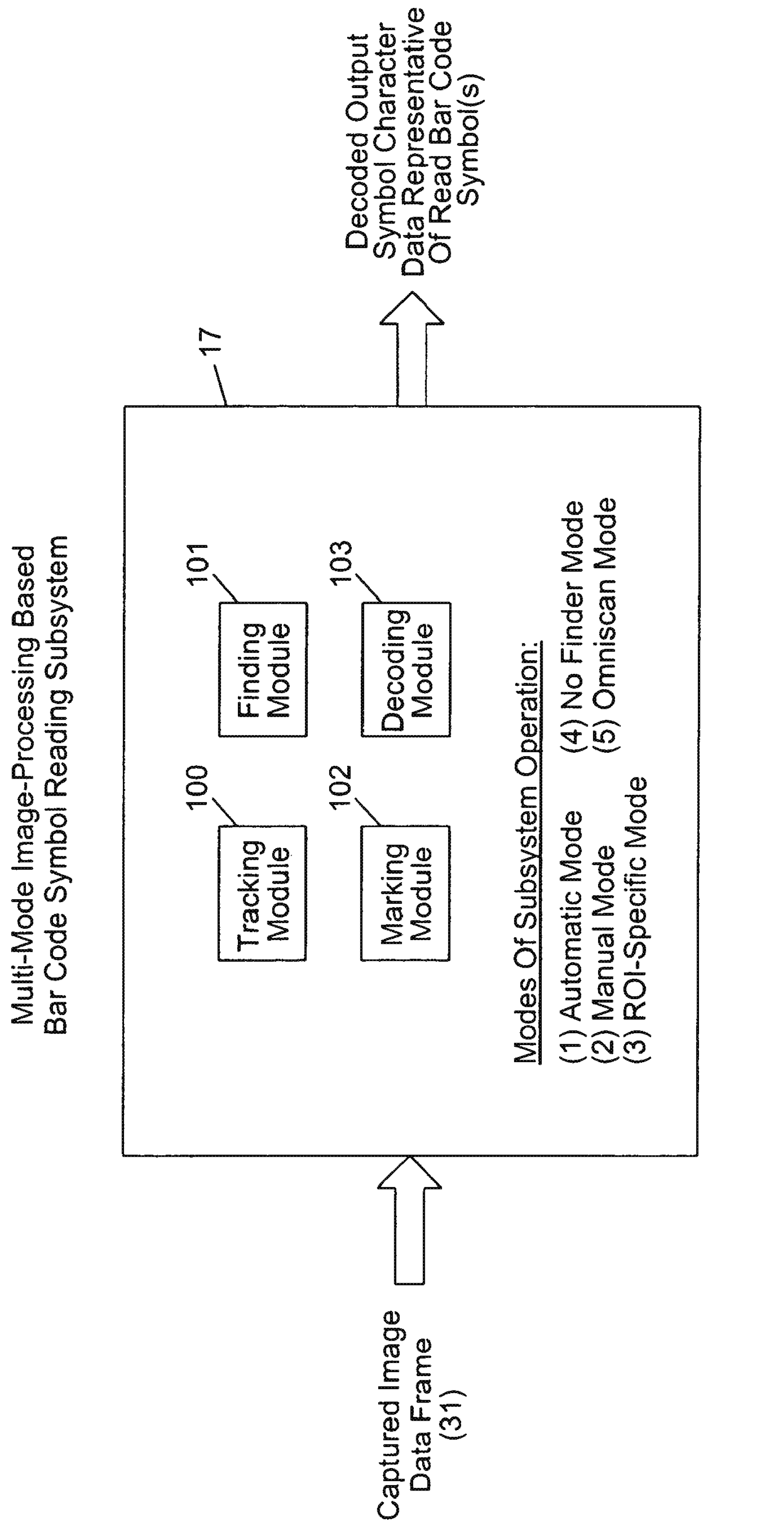

Multi-Mode Image-Processing Based Bar Code Symbol Reading Subsystem
17
Decoded Output Symbol Character Data Representative Of Read Bar Code Symbol(s)
100
Tracking Module
101
Finding Module
102
Marking Module
103
Decoding Module
Modes Of Subsystem Operation:
(1) Automatic Mode
(2) Manual Mode
(3) ROI-Specific Mode
(4) No Finder Mode
(5) Omniscan Mode
Captured Image Data Frame (31)

FIG. 2A2

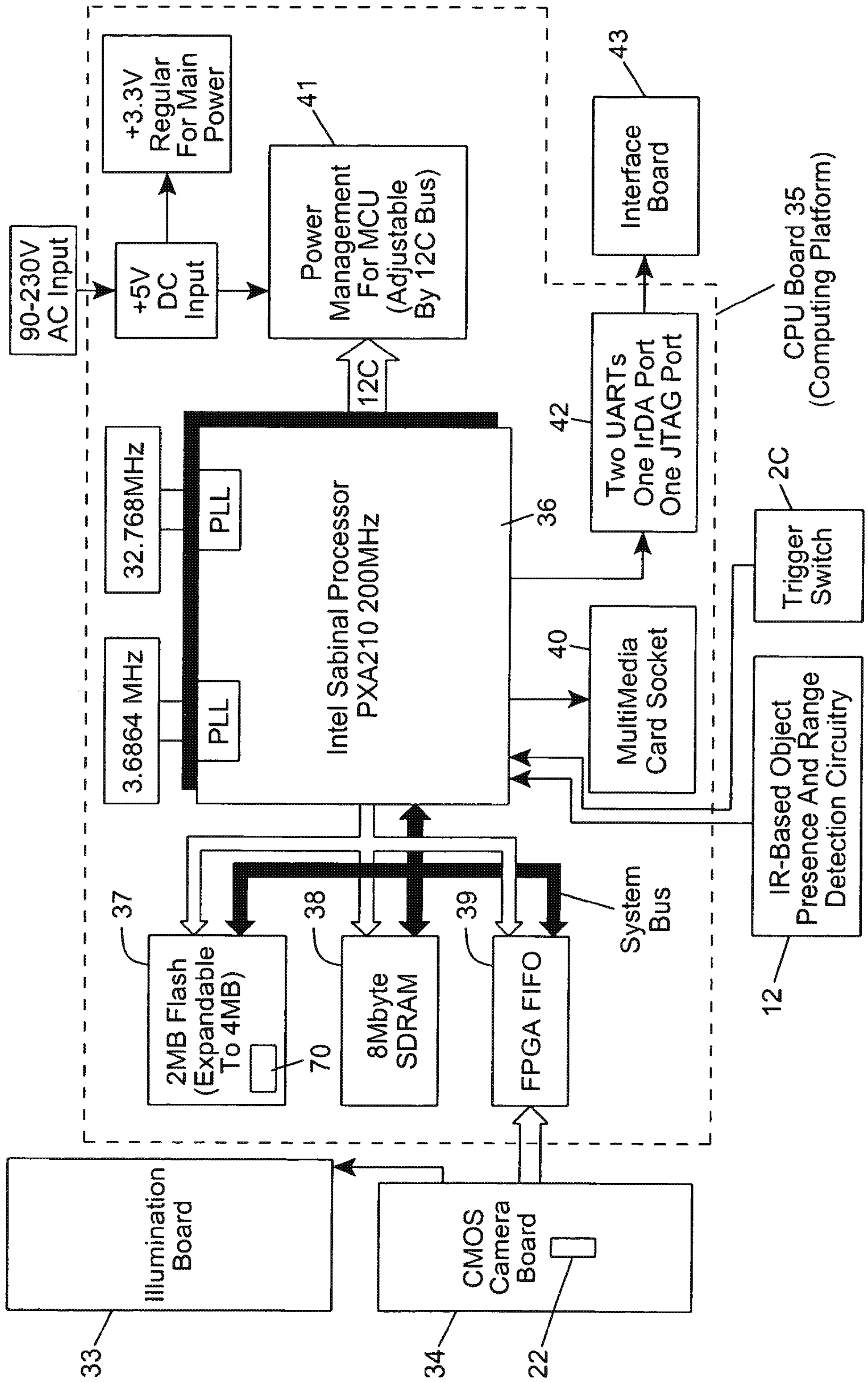

90-230V AC Input
+5V DC Input
+3.3V Regular For Main Power
Power Management For MCU (Adjustable By 12C Bus)
41
12C
43
Interface Board
CPU Board 35 (Computing Platform)
42
Two UARTs One IrDA Port One JTAG Port
32.768MHz
PLL
3.6864 MHz
PLL
Intel Sabinal Processor PXA210 200MHz
36
2C
Trigger Switch
40
MultiMedia Card Socket
IR-Based Object Presence And Range Detection Circuitry
12
37
2MB Flash (Expandable To 4MB)
70
38
8Mbyte SDRAM
39
FPGA FIFO
System Bus
Illumination Board
33
CMOS Camera Board
34
22

FIG. 2B

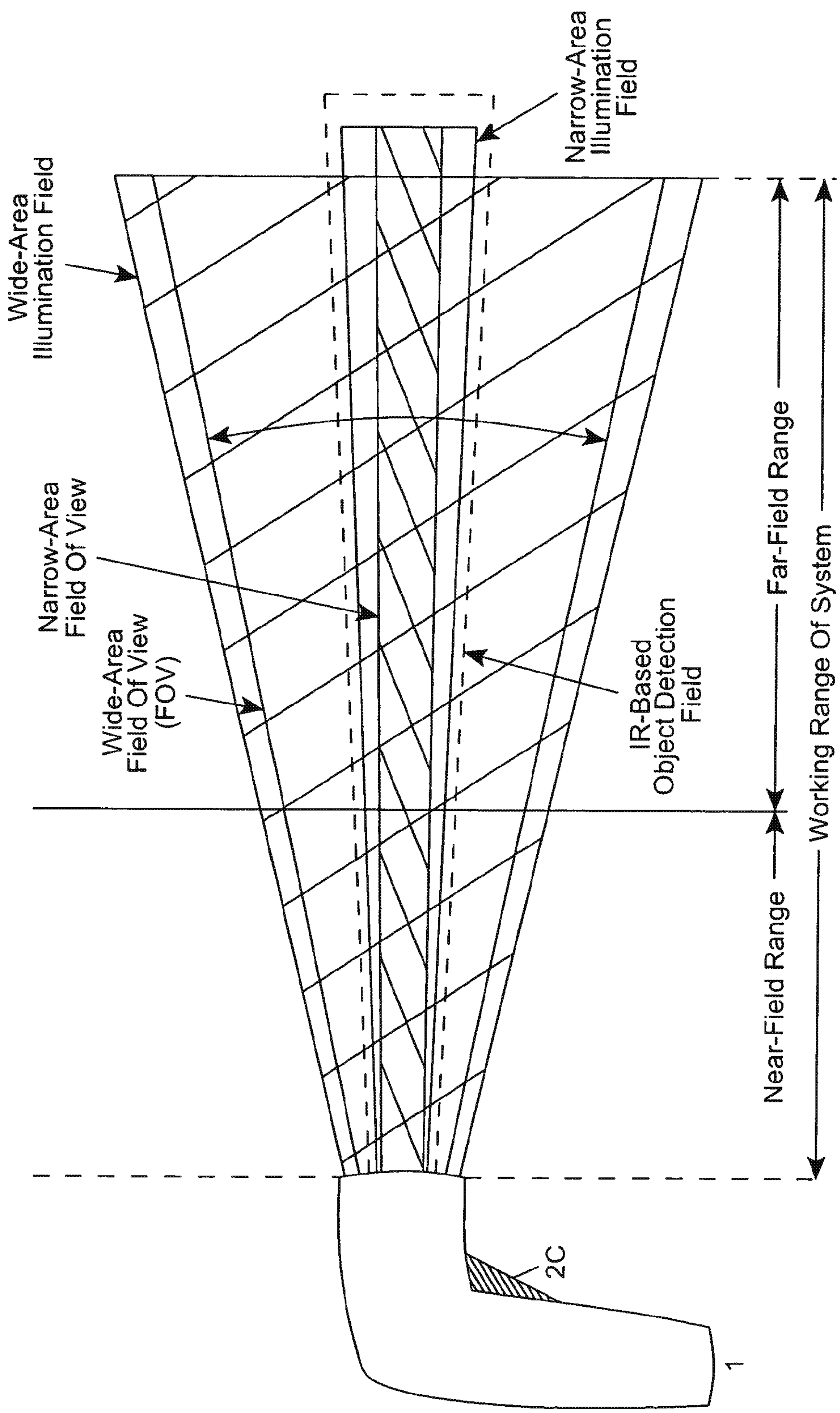

Narrow-Area Illumination Field
Wide-Area Illumination Field
Narrow-Area Field Of View
Wide-Area Field Of View (FOV)
IR-Based Object Detection Field
Far-Field Range
Near-Field Range
Working Range Of System
2C
1

FIG. 3A

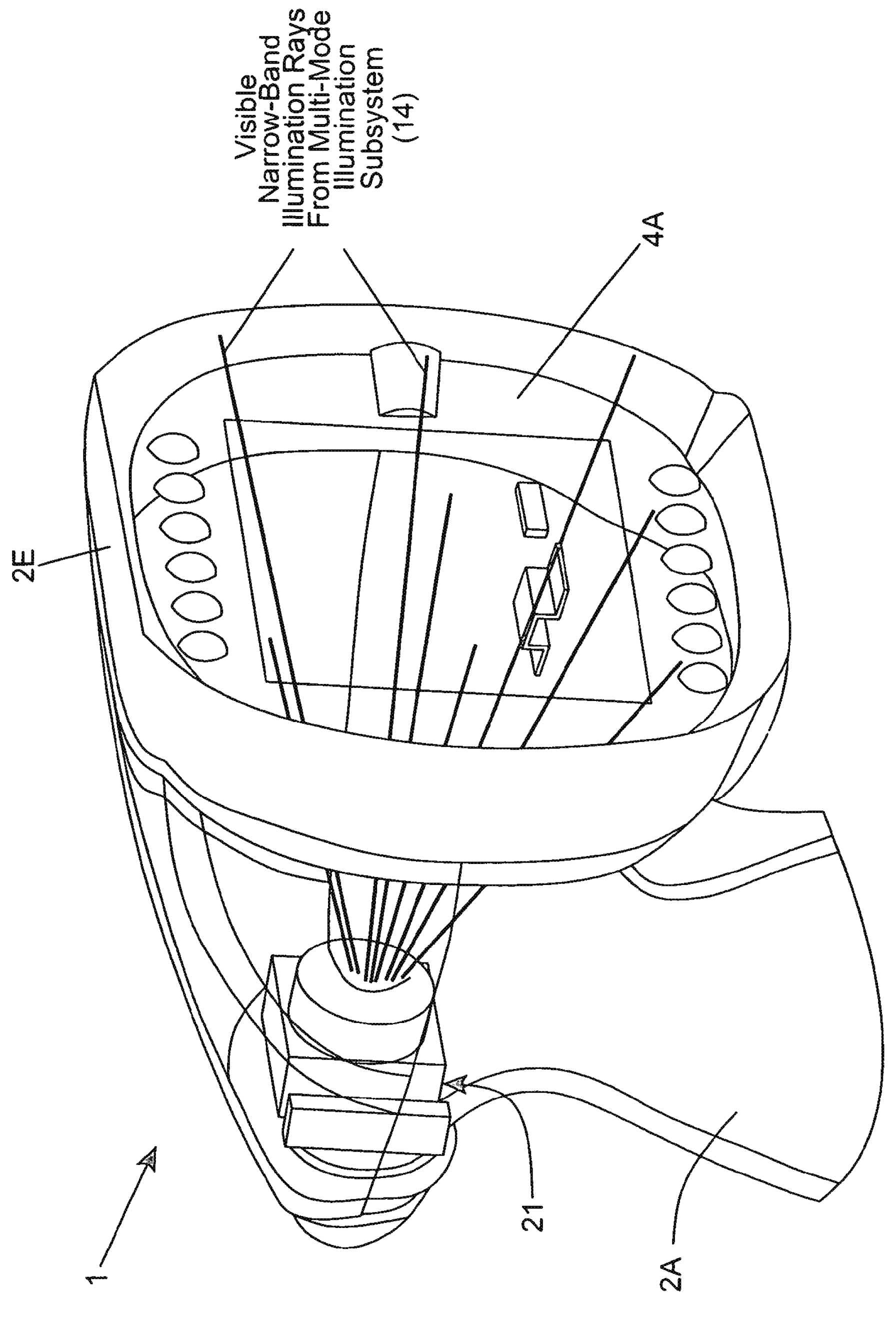

Visible
Narrow-Band
Illumination Rays
From Multi-Mode
Illumination
Subsystem
(14)
4A
2E
1
21
2A

FIG. 3B

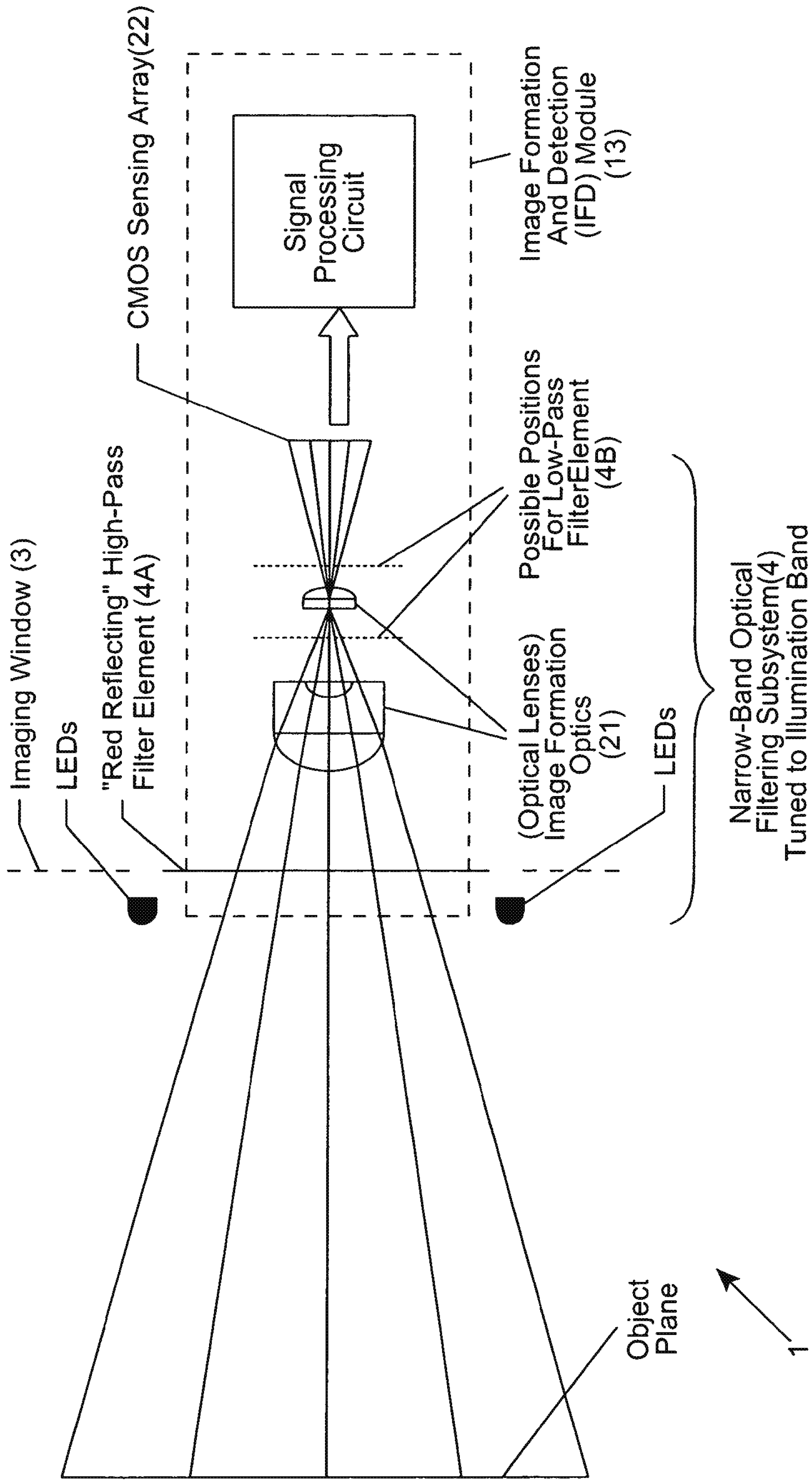

CMOS Sensing Array(22)
Signal Processing Circuit
Image Formation And Detection (IFD) Module (13)
Possible Positions For Low-Pass FilterElement (4B)
Imaging Window (3)
LEDs
"Red Reflecting" High-Pass Filter Element (4A)
(Optical Lenses) Image Formation Optics (21)
LEDs
Narrow-Band Optical Filtering Subsystem(4) Tuned to Illumination Band
Object Plane
1

FIG. 3C

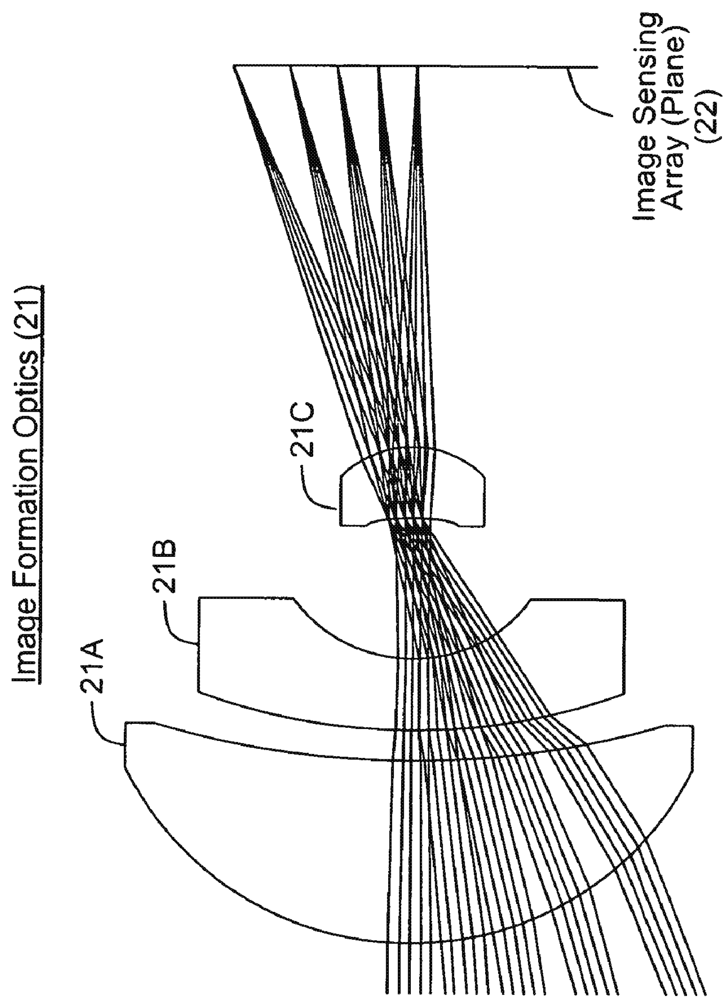


FIG. 3D

- 45° FOV ✓
- As Few Elements As Possible ✓
  - Previous Designs Had 4 Or 5
- As Small As Possible ✓
  - Max Diameter = 12mm
- All Spherical Surfaces ✓
- Common Glasses ✓
  - LaK2 (≈LaK9)
  - ZF10 (=SF8)
  - LaF2 (≈LaF3)

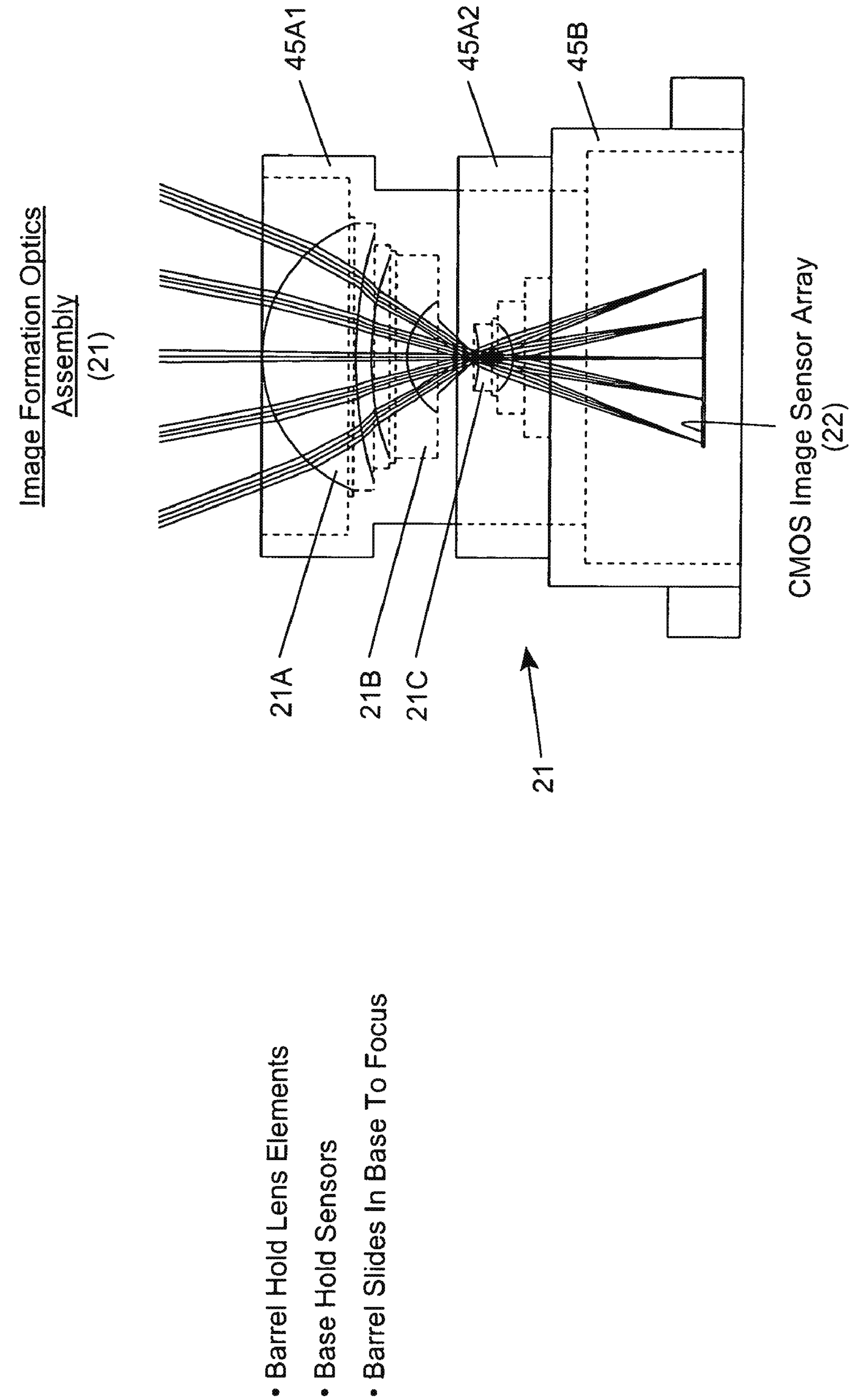


Image Formation Optics Assembly (21)
45A1
45A2
45B
CMOS Image Sensor Array (22)
21A
21B
21C
21
• Barrel Hold Lens Elements
• Base Hold Sensors
• Barrel Slides In Base To Focus

FIG. 3E

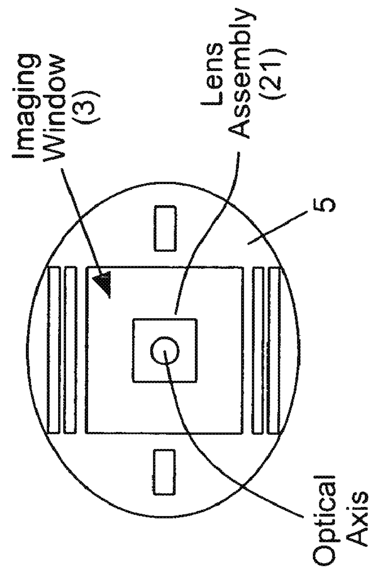


FIG. 3F2

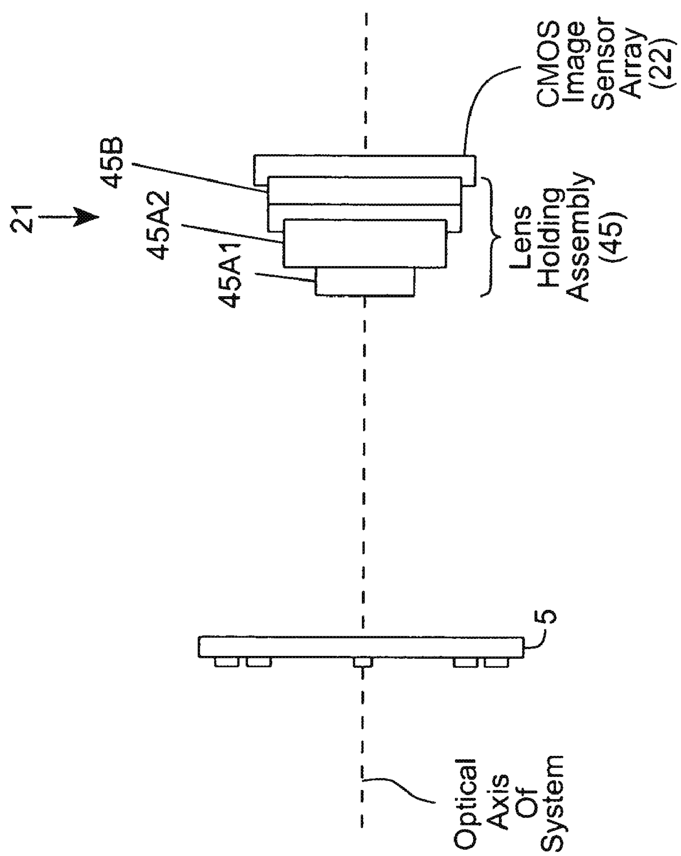
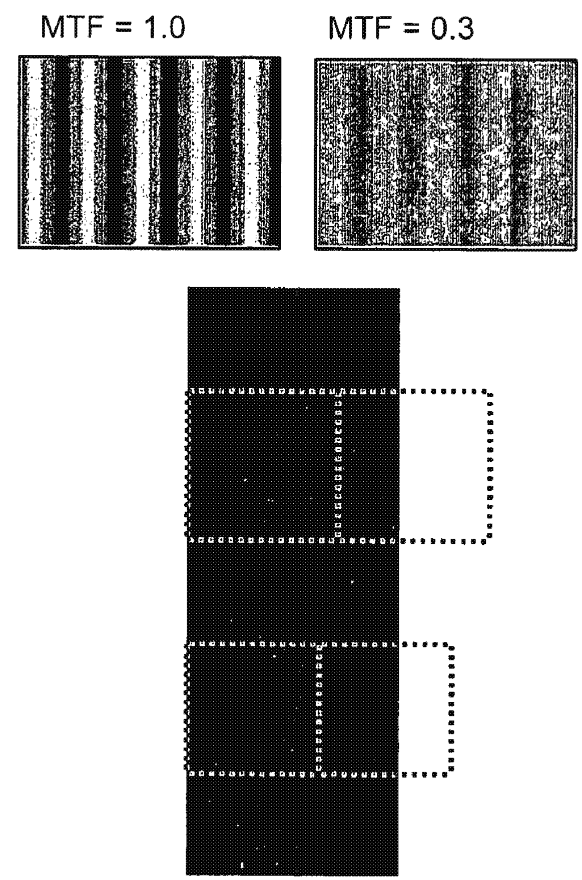


FIG. 3F1

DOF Determination Of Image Formation Optics



- At each distance, find frequency where MTF drops to 0.3
  - Rule of thumb for bar code decoding
  - Depends on code, speed, etc, etc - must test

- BUT: limited by sampling requirement
  - Software needs ~1.6 pixels on narrow code element
  - Limits decode ability regardless of optics
  - Exact value is rule of thumb and flexible (1.4 – 1.6)

FIG. 3G

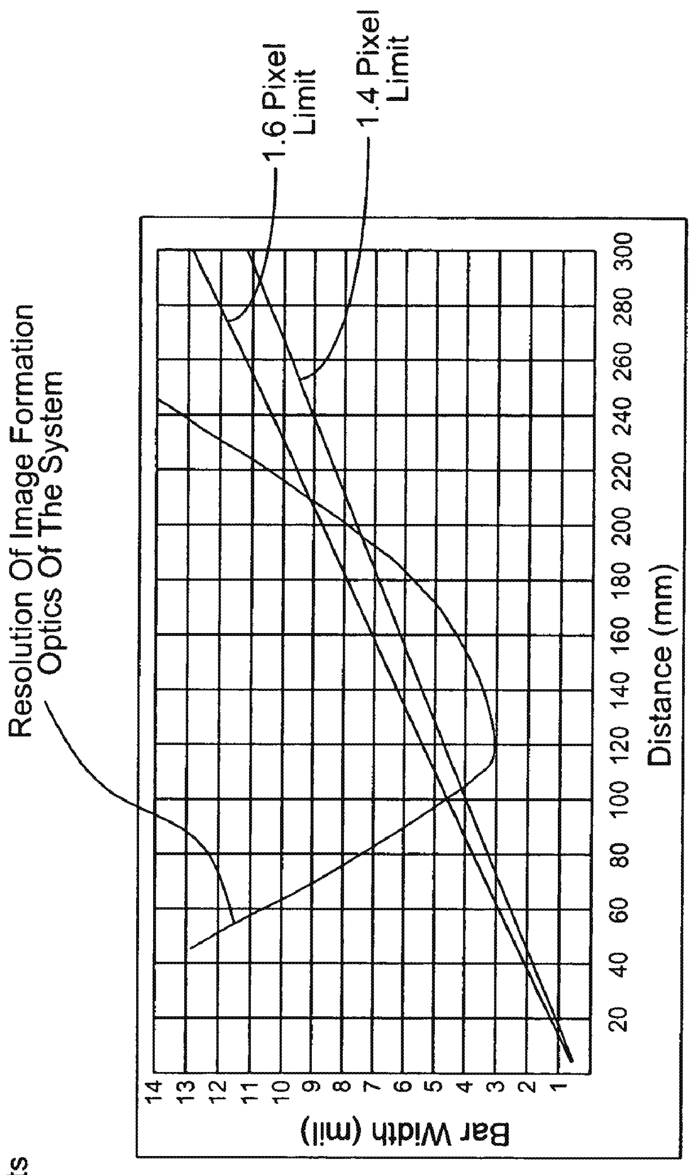


FIG. 4A

Depth Of Field

- Face To 8" For 13.5 Mil
- Optics Resolve 4 Mil Somewhere
- Decodes 5 Mil Somewhere
- No Moving Elements

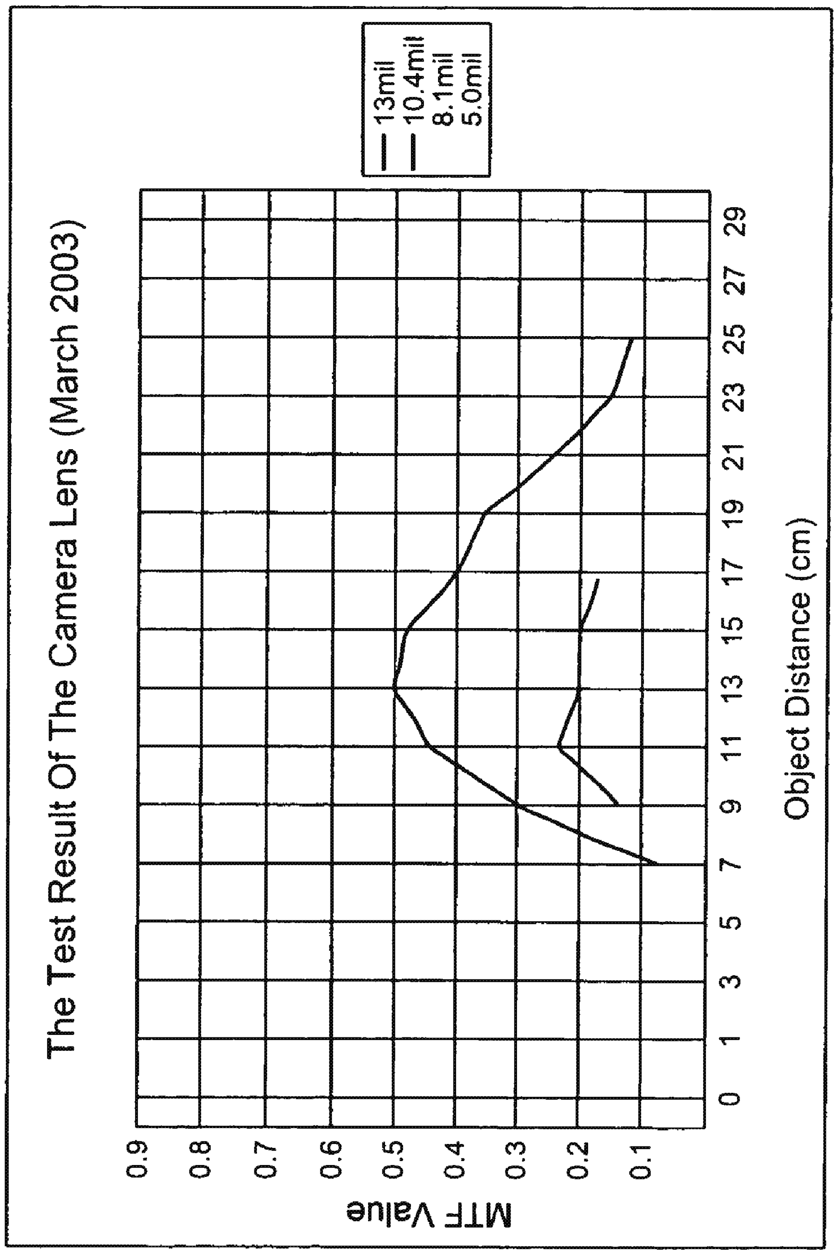
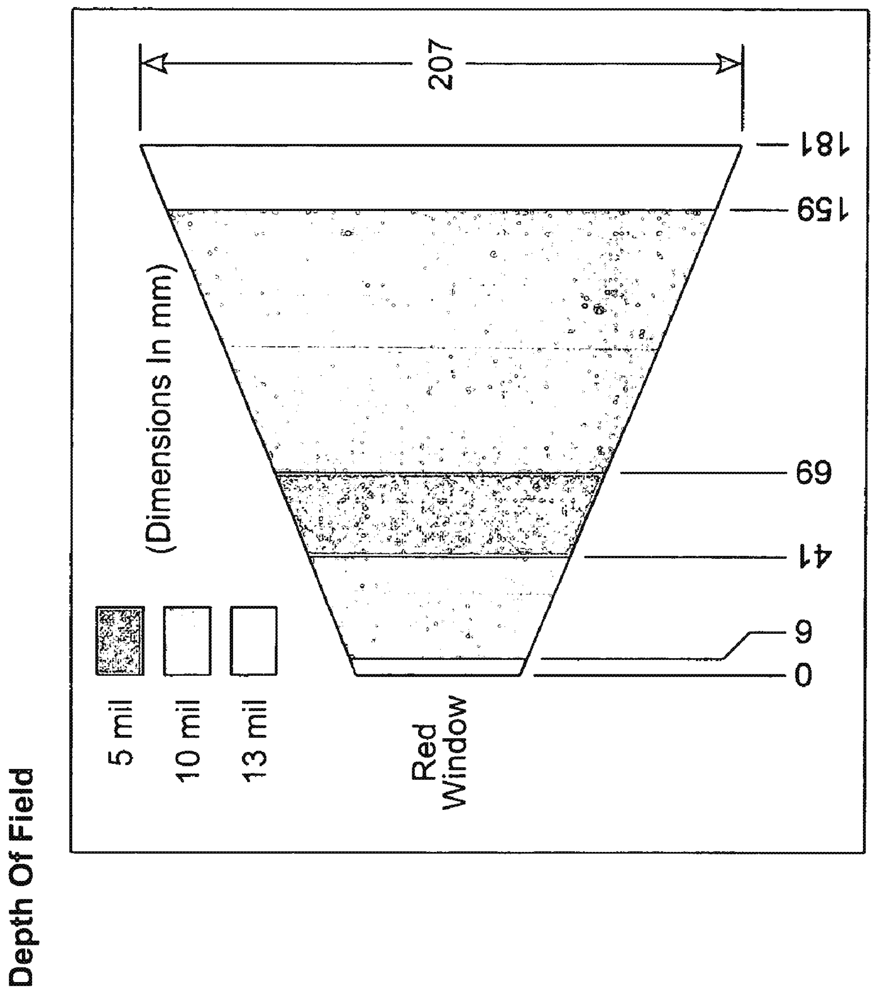
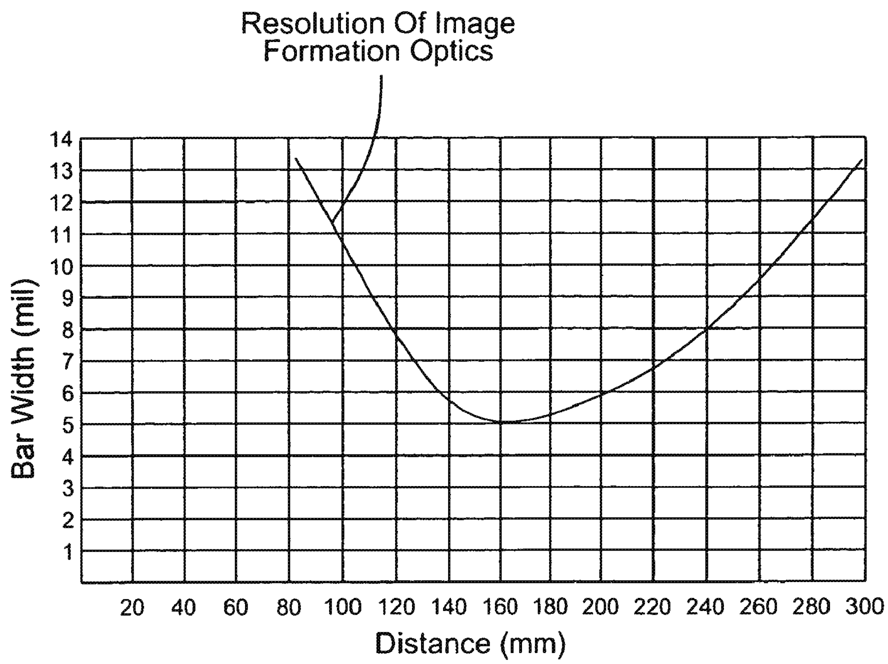

The Test Result Of The Camera Lens (March 2003)
13mil
10.4mil
8.1mil
5.0mil
MTF Value
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Object Distance (cm)
0
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29

FIG. 4B



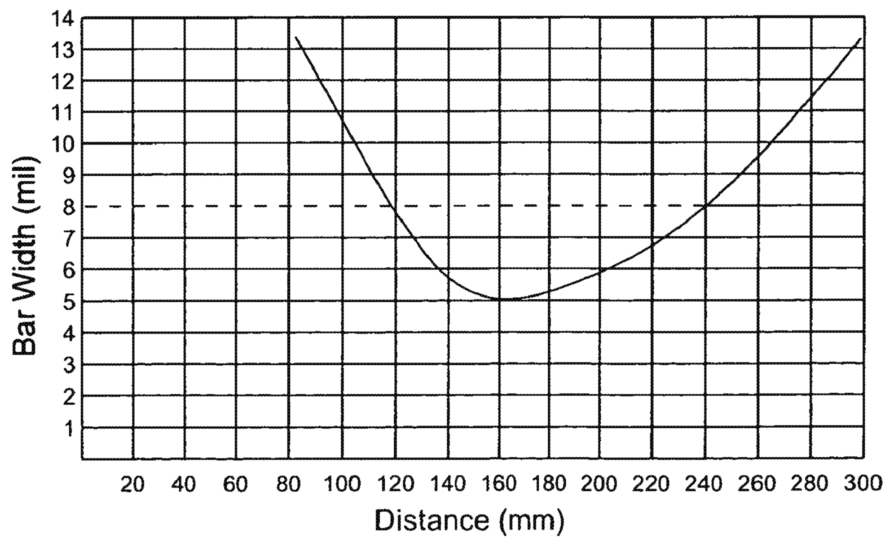
Depth Of Field
(Dimensions In mm)
207
181
159
69
41
9
0
5 mil
10 mil
13 mil
Red Window

FIG. 4C



Resolution Of Image Formation Optics
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Bar Width (mil)
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300
Distance (mm)

FIG. 4D



14
13
12
11
10
9
8
7
6
5
4
3
2
1
Bar Width (mil)
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300
Distance (mm)

FIG. 4E

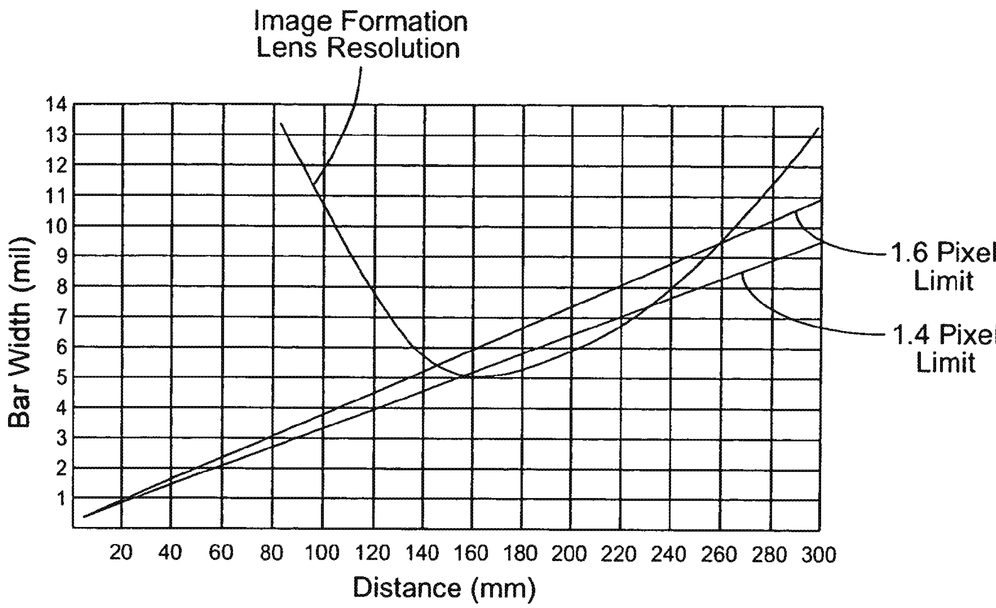


Image Formation
Lens Resolution
1.6 Pixel
Limit
1.4 Pixel
Limit
Bar Width (mil)
14
13
12
11
10
9
8
7
6
5
4
3
2
1
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300
Distance (mm)

FIG. 4F

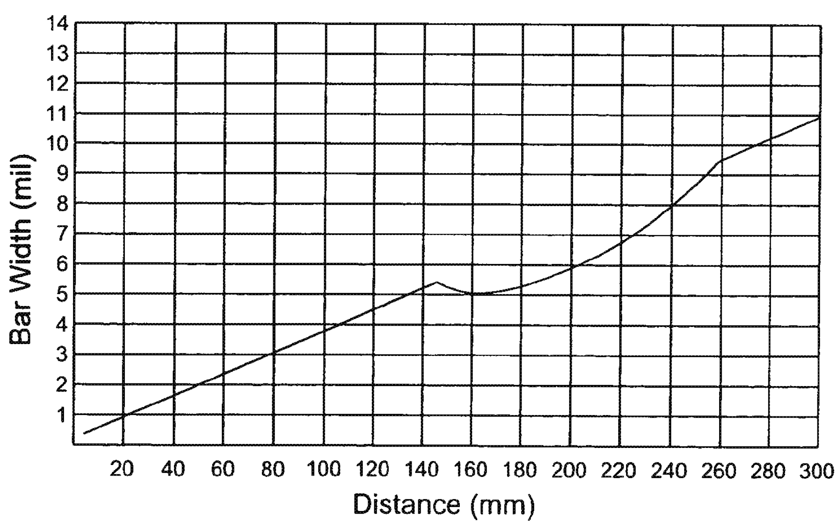
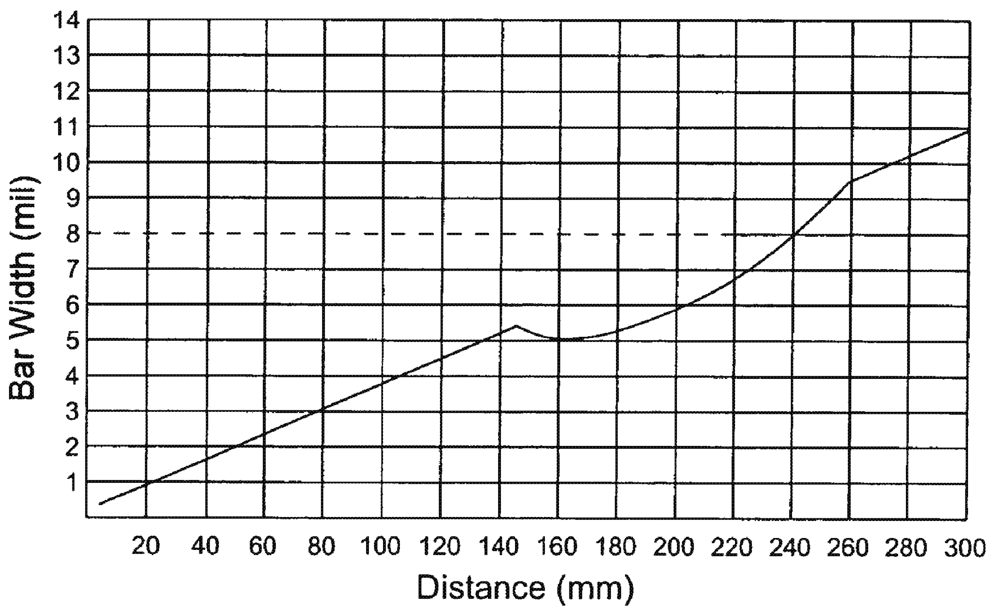

Bar Width (mil)
14
13
12
11
10
9
8
7
6
5
4
3
2
1
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300
Distance (mm)

FIG. 4G



14
13
12
11
10
9
8
7
6
5
4
3
2
1
Bar Width (mil)
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
Distance (mm)

FIG. 4H

DOF_PMAG.zpl

```
graphics
xmx=xmax()
xmn=xmin()
ymx=ymax()
ymn=ymin()
xwidth=xmx-xmn
ywidth=ymx-ymn
xleft=xmn+(0.l*xwidth)
xrigh=xmn+(0.95*xwidth)
ytopp=ymn+(0.05*ywidth)
ybott=ymn+(0.7*ywidth)

line xleft,ytopp,xrigh,ytopp
line xrigh,ytopp,xrigh,ybott
line xrigh,ybott,xleft,ybott
line xleft,ybott,xleft,ytopp

format 4.3
settextsize 140,80
gtext 0.68*xwidth,(0.85)*ywidth,0,"Wav : "
gtext 0.68*xwidth, (0.88)*ywidth,0, "WGT : "
for i=l,nwav(),1
gtext (0.68+i*0.05)*xwidth,0.85*ywidth,0,$str(wavl(i))
gtext (0.68+i*0.05)*xwidth,0.88*ywidth,0,$str(wwgt(i))
next
gtext 0.68*xwidth,(0.91)*ywidth,0,"Relative illumination: "
gtext 0.9*xwidth,(0.91)*ywidth,0,$str(reli(nfld()))
settextsize 90,50
input "Please input startpoint (mm):",start
if (start<=0) then input "Please input startpoint (mm):", start
input "Please input pixel size (um):",pix
if (pix<=O) then input "Please input pixel size (um):",pix
for i=start,start+150,10
xpos=xleft+(i-start)/150*0.85*xwidth
line xpos,ytopp,xpos,ybott
format3.0
gtext xleft*0.85+(i-start)/150*0.85*xwidth,0.72*ywidth,0,$str(i)
next
settextsize 70,40
for i=1,14,1
ypos=ytopp+i/14*.65*ywidth
line xleft,ypos,xrigh,ypos
format 3.0
gtext 0.05*xwidth,ytopp*0.9+(j-1)/14*.65*ywidth,0,$str(14-i+1)
next

gtitle "The DOF and PMAG curve of current design"
gdate

format 12.6
oldthic=thic(0)

getsystemdata 2
settextsize 120,40
j=1
gtext xwidth*0.018,0.85*ywidth,0,"centering "
for i=1,nsur()-2,1
        if (gind(i)!=0.0)
                format 2.0
                gtext xwidth*0.10+(j-1)*0.07*xwidth,0.85*ywidth ,0,$str(j)+":"
                gtext xwidth*0.12+(j-1)*0.07*xwidth,0.85*ywidth ,0,":"
                format 4.2
```

FIG. 4I1

DOF_PMAG.zpl

```
                    if(curv(i)*curv(i+1)<0) then
centering=abso((sdia(i)*curv(i)+sdia(i+1)*curv(i+1)))
                    if(curv(i)*curv(i+1)>0) then
centering=abso((sdia(i)*curv(i)-sdia(i+1)*curv(i+1)))
                    gtext xwidth*0.13+(j-1)*0.07*xwidth,0.85ywidth,0,$str(centering)
                    j=j+1
        endif
next
format 4.2
settextsize 70,40
gtext xwidth*0.018,0.91*ywidth,0,"image space f/# : "+$str(vec2(8))
gtext xwidth*0.018,0.94*ywidth,0,"effective focal length: "+ $str(vec2(7))
!color (3)
gtextcent ymn+(0.77*ywidth),"distance (mm)"
gtext xleft*0.32,0.5*ywidth,90,"bar width (mil)"

format 12.6
settextsize 100,40
minmtf=1
maxfreq=0
thic 0=start
update all
for k=0,200,0.2
          !i=nfld()
            for i=1,nfld(),1
                      getmtf k,O,i,2,1,1
                      !print vec1(0)
                      !print vec1(1)
                      if (vec1(0)<minmtf) then minmtf=vec1(0)
                      if (vec1(1)<minmtf) then minmtf=vec1(1)
                      if (minmtf<=0.3)
                                maxfreq=k
                                goto 1
                      endif
            next
next
label 1
!color (1)

!output "1.txt" append

oldxpos=xleft+0/150*0.85*xwidth
oldypos=ytopp+(14-(1/(maxfreq/(sdia(0)/sdia(nsur())))*0.5/25.4*1000))/14*0.65*ywidth
switch=0
m=0
for j=start,start+150,3
            thic 0=j
          update all
            minmtf=1
            for k=m,200,0.3
                      !i=nfld()
                      for i=1,nfld(),1
                      getmtf k,0,i,2,1,1
                      if (vec1(0)<minmtf) then minmtf=vec1(0)
                      if (vec1(1)<minmtf) then minmtf=vec1(1)
                      if (minmtf<=0.3)
                                maxfreq=k
                                goto 2
                      endif
                      next
            next
            label 2
            if (maxfreq-5)>0
```

FIG. 4I2

HONEYWELL-00192875

DOF_PMAG.zpl

```
                m=maxfreq-10
        else
                m=0
        endif
        !print j,sdia(0),sdia(nsur()),maxfreq
        if ((switch==0) & (1/(maxfreq/(sdia(0)/sdia(nsur()))) 0.5/25.4*1000<=13))
                !color (0)
                format 5.2
                a$="FOV for 10 mil: "+$str(2*sdia(0)) + " at "+$str(j-2)+ mm ; "
                gtext xwidth*0.018,0.97*ywidth,0,a$
                switch=1
                format 12.6
                !color(1)
        else
                        if ((switch==1) &
(1/(maxfreq/(sdia(0)/sdia(nsur())))*0.5/25.4*1000>=13))
                            !color(0)
                            format 5.2
                            a$=$str(2*sdia(0))+" at "+$str(j-2)+" mm"
                            gtext xwidth*0.44,0.97*ywidth,0,a$
                            switch=0
                            format 12.6
                            goto 3
                            !color(1)
                        endif
                endif
                newxpos=        xleft+(j-start)/150*0.85*xwidth

newypos=ytopp+(14-(1/(maxfreq/(sdia(0)/sdia(nsur())))*0.5/25.4*1000))/14*0.65*ywidth
                if ((14-14*(oldypos-ytopp)/0.65/ywidth)<14) then line
oldxpos,oldypos,newxpos,newypos
                oldxpos=newxpos
                oldypos=newypos
next
label 3
thic 0=start
update all
oldxpos=xleft+0/150*0.85*xwidth
oldxpos1=xleft+0/150*0.85*xwidth
oldypos=ytopp+(14-(0.5/((0.5/1.6/pix*1000)/(sdia(0)/sdia(nsur())))/25.4*1000))/14*0.
65*ywidth
oldypos1=ytopp+(14-(0.5/((0.5/1.4/pix*1000)/(sdia(0)/sdia(nsur())))/25.4*1000))/14*0
.65*ywidth
for j=start,start+150,4
                thic 0=j
                update all
                newxpos=xleft+(j-start)/150*0.85*xwidth
                newxpos1=xleft+(j-start)/150*0.85*xwidth

newypos=ytopp+(14-(0.5/((0.5/1.6/pix*1000)/(sdia(0)/sdia(nsur())))/25.4*I000))/14*0.
65*ywidth

newypos1=ytopp+(14-(0.5/((0.5/1.4/pix*1000)/(sdia(0)/sdia(nsur())))/25.4*1000))/14*0
.65*ywidth
            line oldxpos, oldypos,newxpos, newypos
            line oldxpos1,oldypos1,newxpos1,newypos1
            oldxpos=newxpos
            oldypos=newypos
            oldxpos1=newxpos1
            oldypos1=newypos1
next

thic 0=oldthic
```

FIG. 4I3

HONEYWELL-00192876

Multi-Mode Illumination Subsystem

- Three Modes Of Illumination
  (1) Wide-Area For "Near" Object (0 mm-100 mm)
  (2) Wide-Area For "Far" Object (100 mm-200 mm)
  (3) Narrow-Area For "Near" Object (30 mm-100 mm)

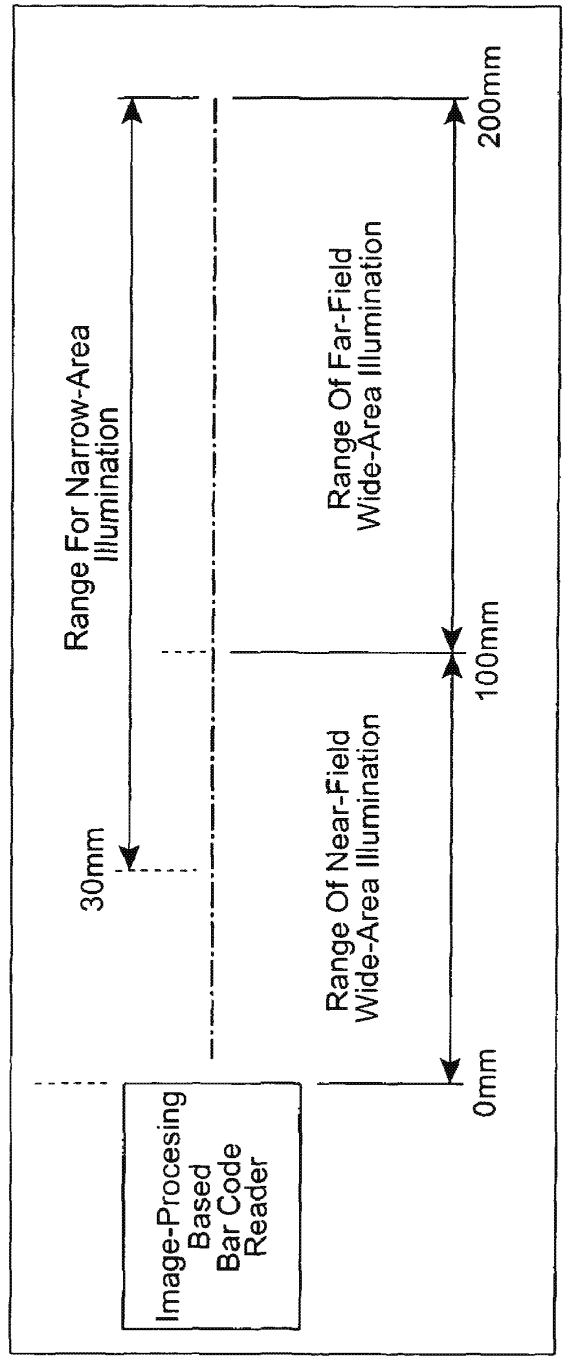


FIG. 5A1

Illumination Design Goals For First Illustrative Embodiment

- Wide-Area Illumination Modes
  - Match FOV and DOF (45°, 200mm)
  - Sufficient power density on target
    - Pixel value > 80 DN at far field center
  - Achieve sufficient uniformity (center:edge = 2:1 max)
  - Use as few LEDs as possible
- Narrow-Area Illumination Mode
  - Line usable beginning 40 mm from window
  - Match FOV and DOF
  - Sufficient power density on target
  - Sufficiently thin line
    - Height < 10 mm at far field

FIG. 5A2

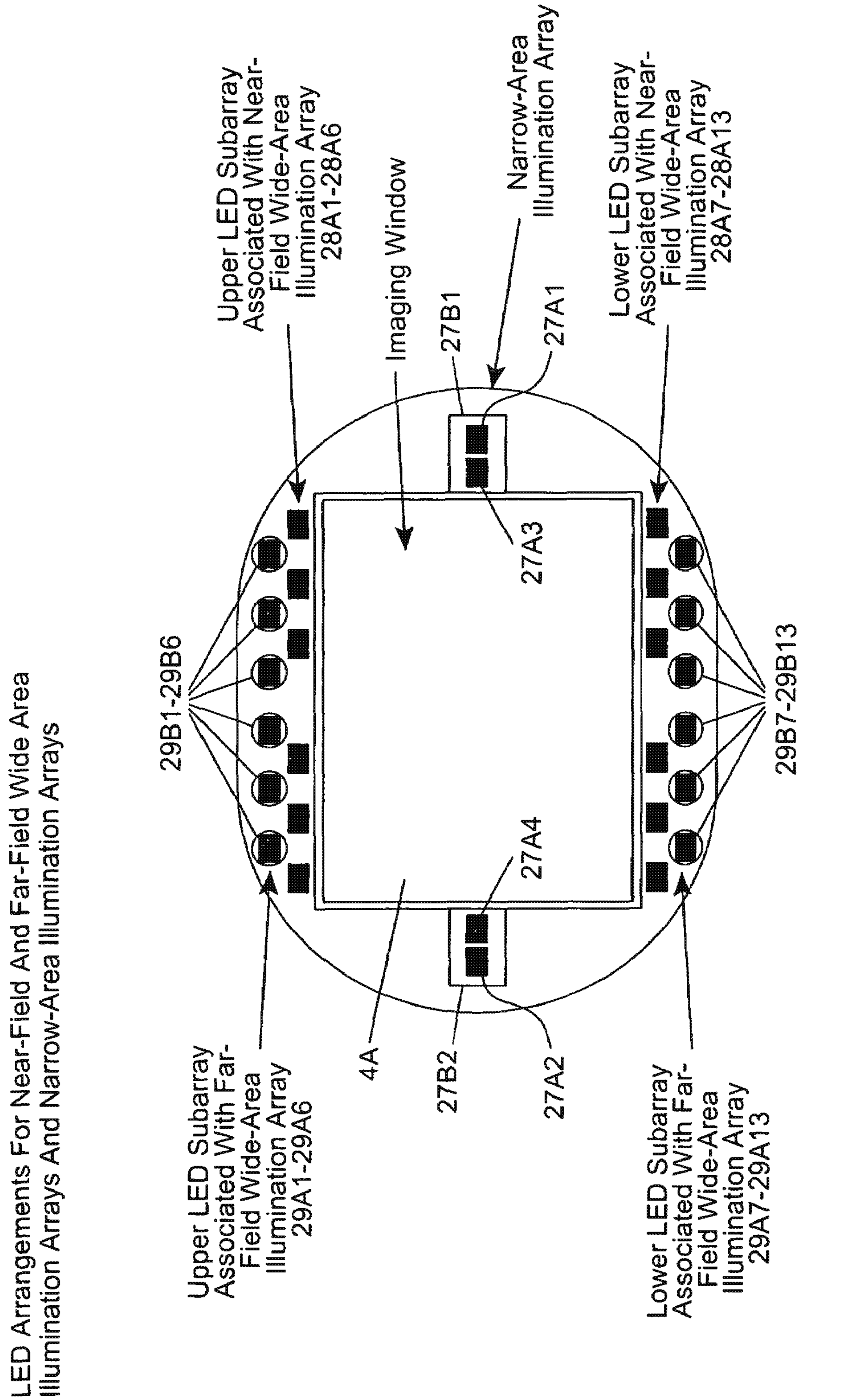
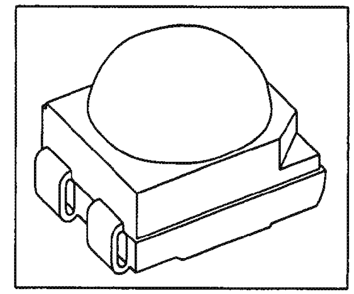

LED Arrangements For Near-Field And Far-Field Wide Area Illumination Arrays And Narrow-Area Illumination Arrays
Upper LED Subarray Associated With Near-Field Wide-Area Illumination Array 28A1-28A6
Imaging Window
27B1
Narrow-Area Illumination Array
27A1
Lower LED Subarray Associated With Near-Field Wide-Area Illumination Array 28A7-28A13
29B1-29B6
27A3
29B7-29B13
27A4
Upper LED Subarray Associated With Far-Field Wide-Area Illumination Array 29A1-29A6
4A
27B2
27A2
Lower LED Subarray Associated With Far-Field Wide-Area Illumination Array 29A7-29A13

FIG. 5B



LEDs For Narrow-Area Illumination

- Linear Illumination: Osram LS E655
  - 633 nm InGaAlP
  - 60° Lambertian Emittance
  - 6.75 mW Total Output Power (Typical Conditions)
  - $0.18 Each In 50k

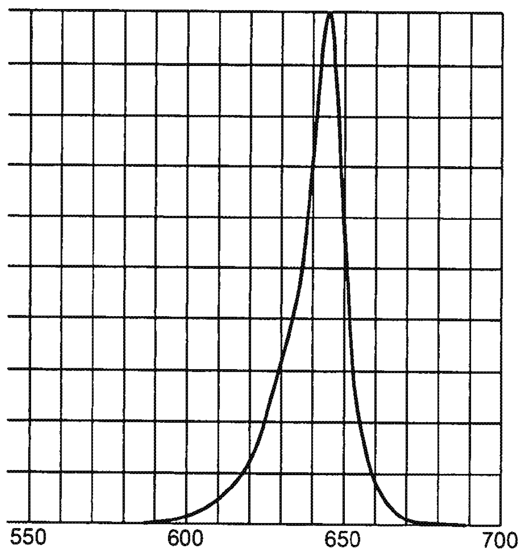


FIG. 5C1

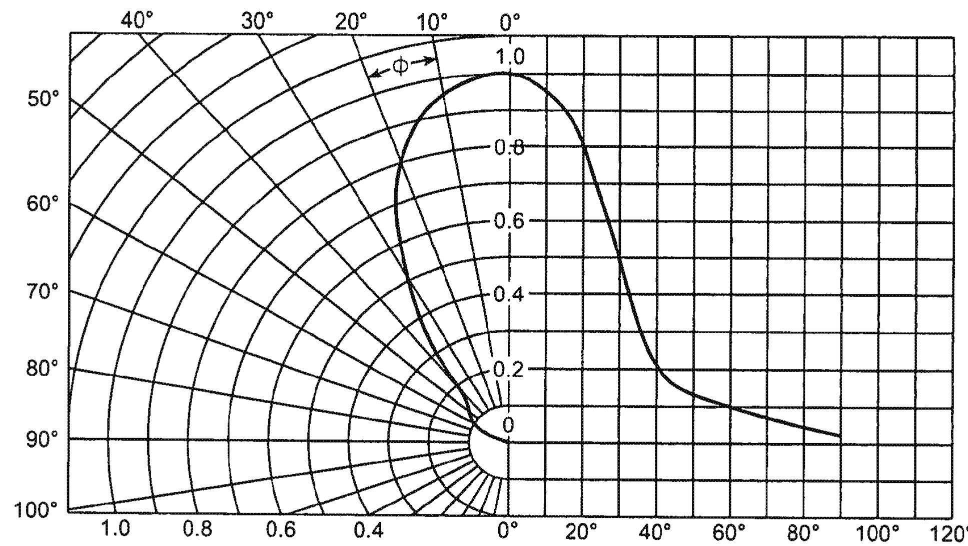


FIG. 5C2

Cylindrical Lenses For Narrow-Area Illumination Array

- First Surface Curved Vertically To Create Line
- Second Surface Curved Horizontally To Control Line Height

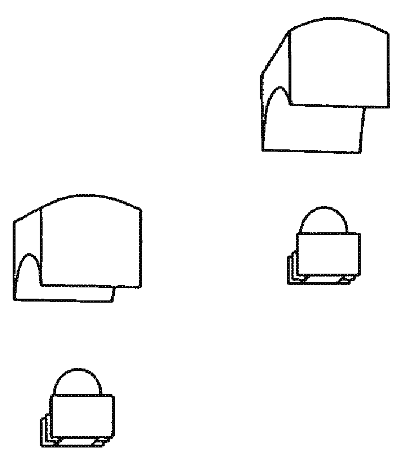


FIG. 5C3

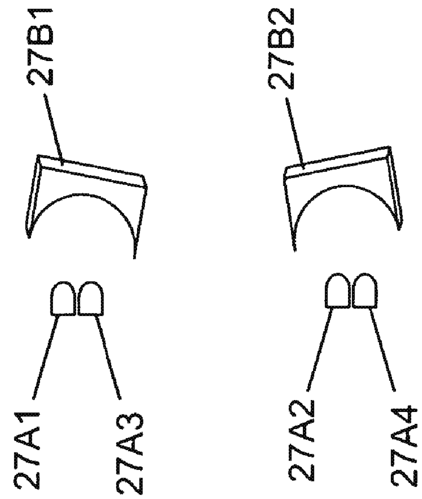


FIG. 5C4

Linear Illumination Profiles

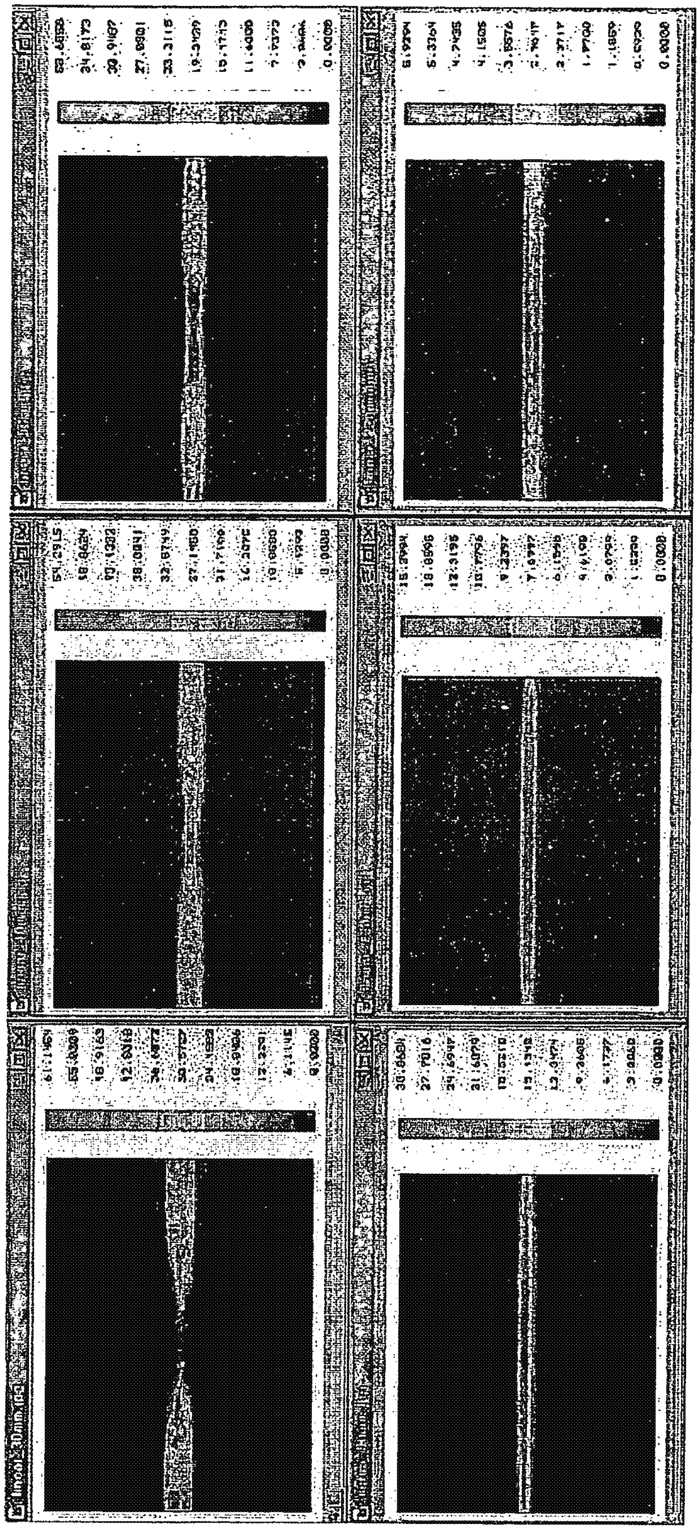
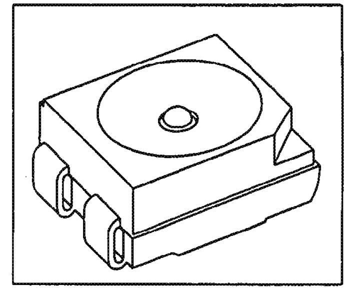


FIG. 5C5

Area LEDs

- Area Illumination: Osram LS E67B
  - 633 nm InGaAlP
  - 120° Lambertian Emittance
  - 11.7 mW Total Output Power (Typical Conditions)
  - $0.18 each In 50k



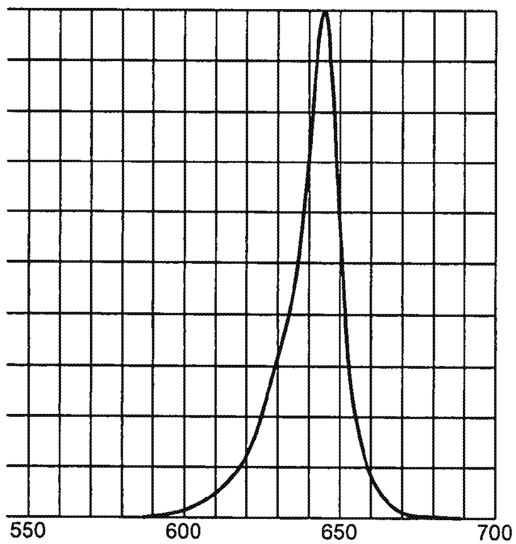


FIG. 5D1

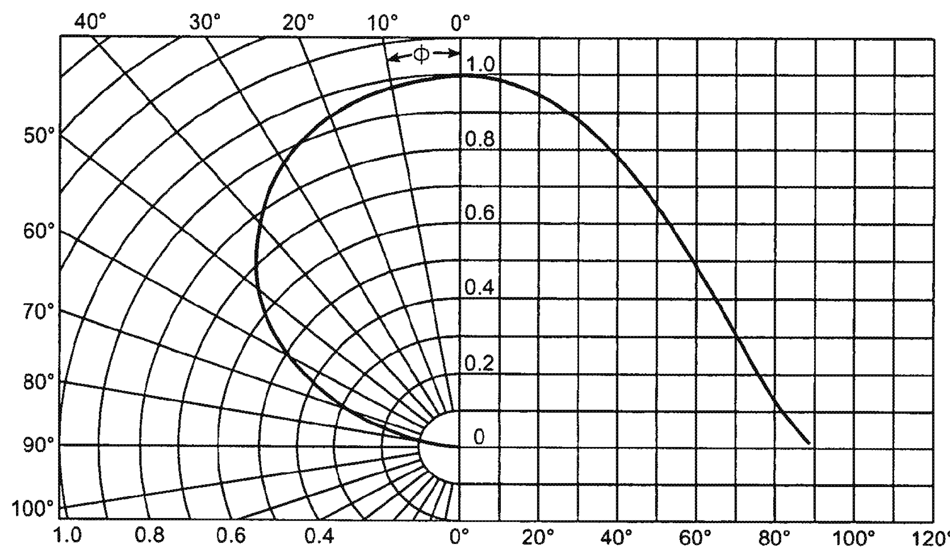


FIG. 5D2

Far Area Lenses

- Plano Convex Lenses In Front Of Far Field LEDs

- Light Aimed By Angling Lenses
  - Even Out Distribution Across FOV Throughout DOF
  - Satisfy Center: Edge = 2:1 Max Criterion
  - Allows LEDs To Be Mounted Flat

- All Lenses CNCed In Single Piece Of Plastic

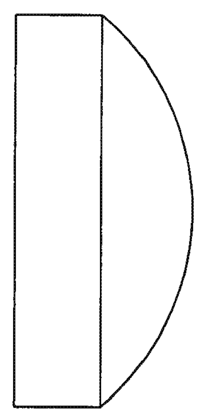


FIG. 5D3

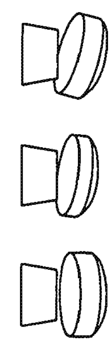


FIG. 5D4

Wide-Area Illumination Profiles (Near)

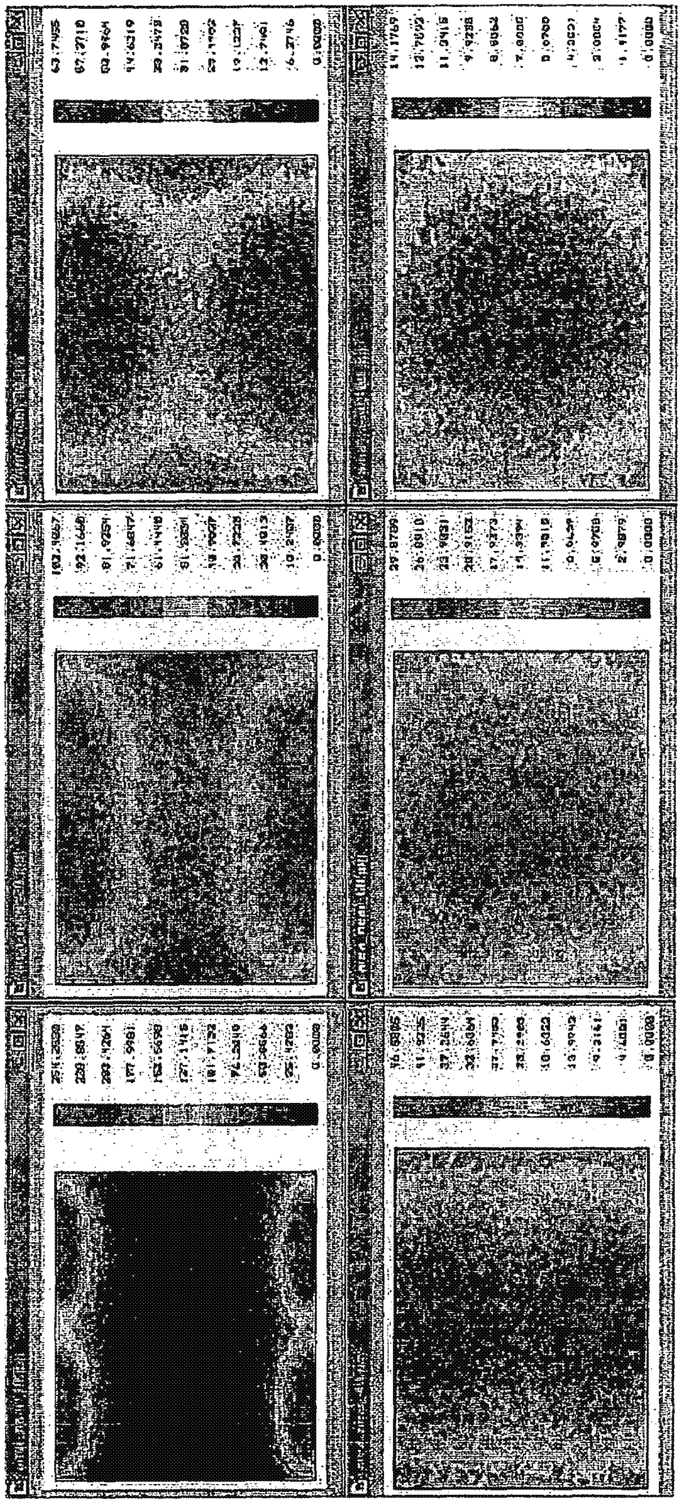


FIG. 5D5

Wide-Area Illumination Profiles (Far)

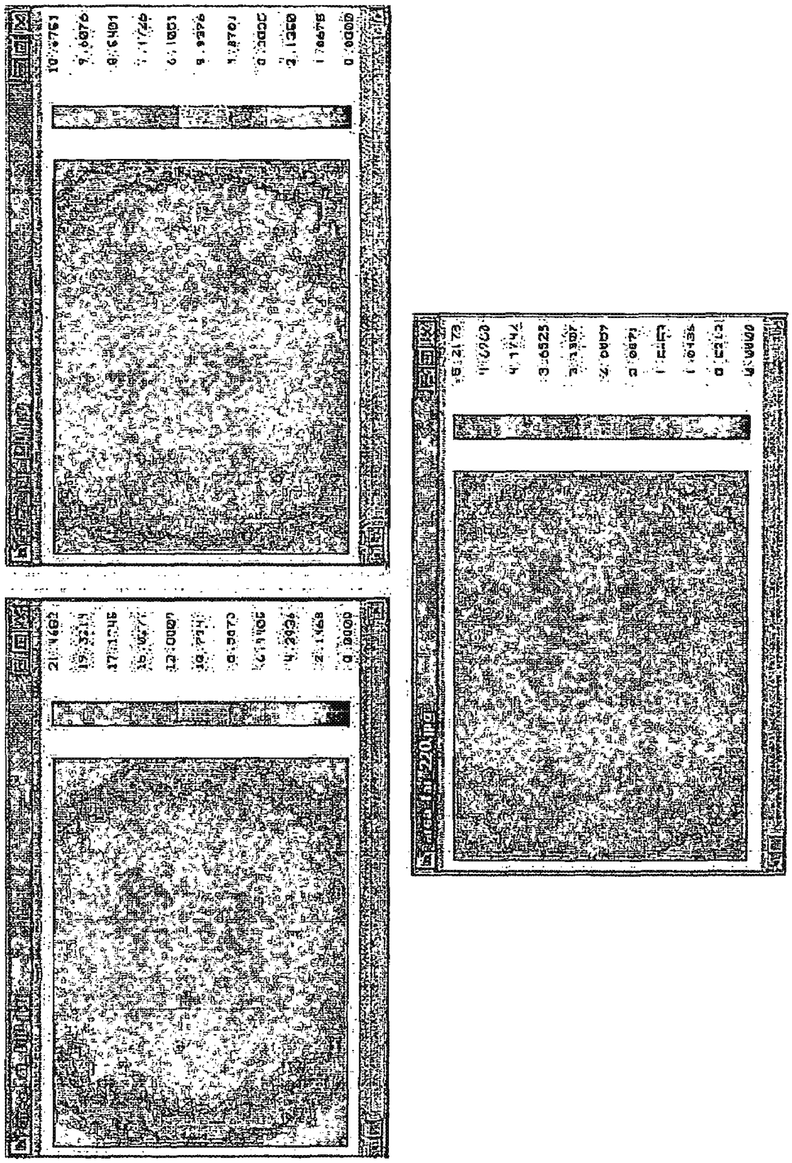


FIG. 5D6

Pixel Value Calculation

- Pixel Value Calculation For Center Of Far-Field Shows Sufficient Signal (> 80DN)

Description		Value	Unit
Sensor Power Density	Target Power Density	4	µW/mm²
	Surface Reflection	0.6	#
	Optical Transmittance	0.9	
	F-Number	9	
	Pixel Power Density	0.007	µW/mm²
Signal	CMOS Internal Gain	4.5	#
	Amplification Gain	20	dB
	Integration Gain	5	ms
	Sensor Responsivity	1.8	V / (lx s)
	Wavelength	633	nm
	Photopic Luminous Efficiency	0.238	lm / W
	Signal Out Of Sensor	0.439	V
Pixel Value	A/D Range Max	1.3	V
	A/D Range Min	0.0	
	Pixel Value (0-255)	86	DN

FIG. 5D7

HONEYWELL-00192887

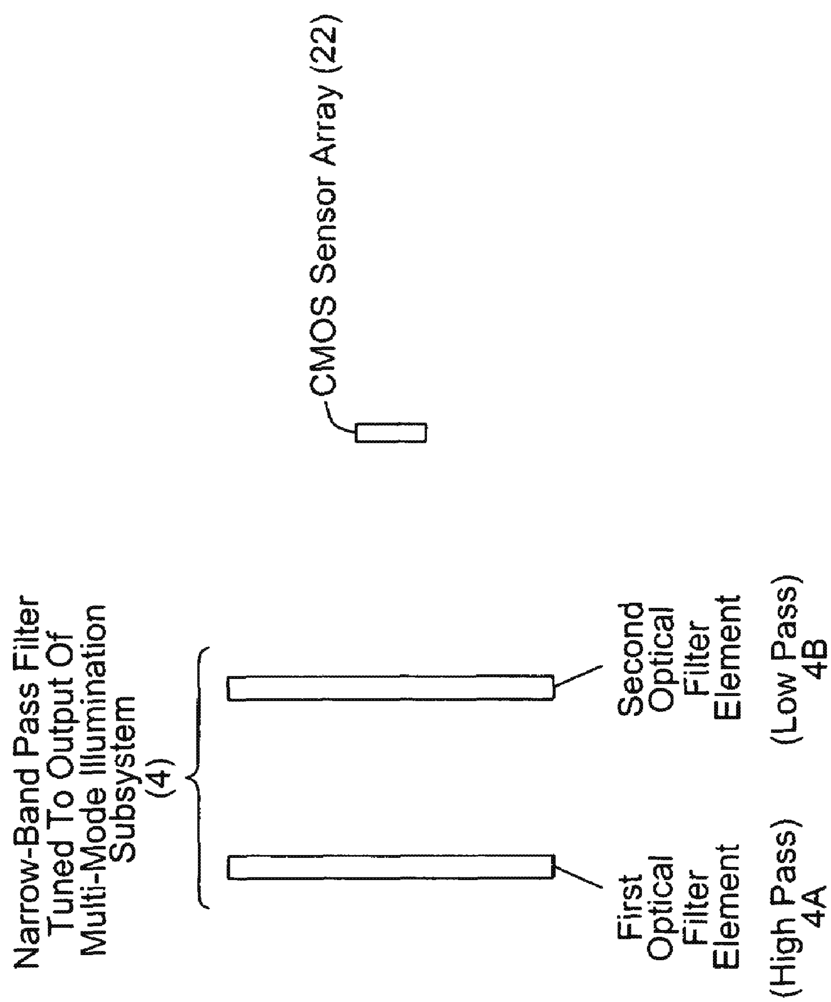

CMOS Sensor Array (22)
Narrow-Band Pass Filter Tuned To Output Of Multi-Mode Illumination Subsystem (4)
Second Optical Filter Element
(Low Pass) 4B
First Optical Filter Element
(High Pass) 4A

FIG. 6A1

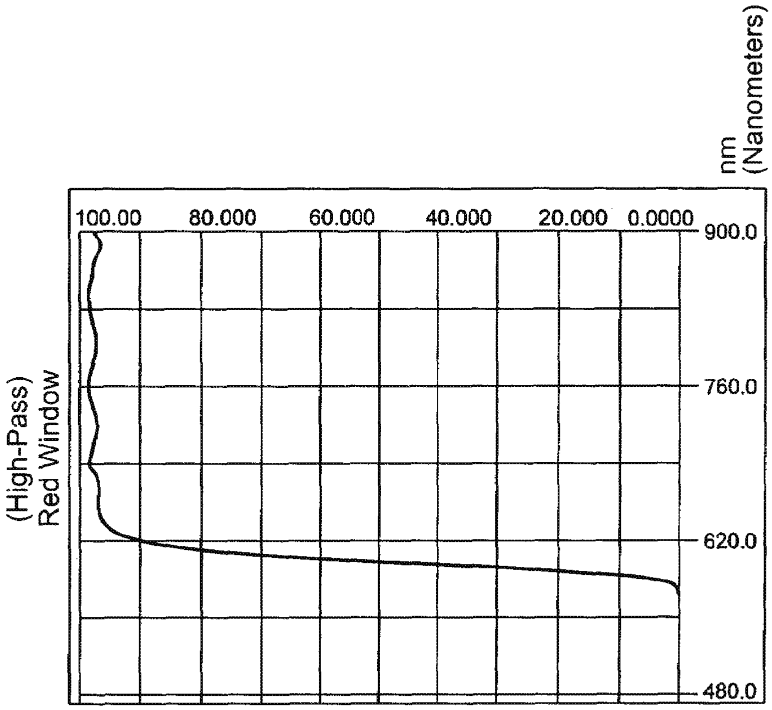


FIG. 6A3

Red Window And Low-Pass Filter Characteristics

- Must Bandpass Return Light Against Ambient
  - Red Window + Low Pass Filter
  - Restricts Range To 620nm – 700nm

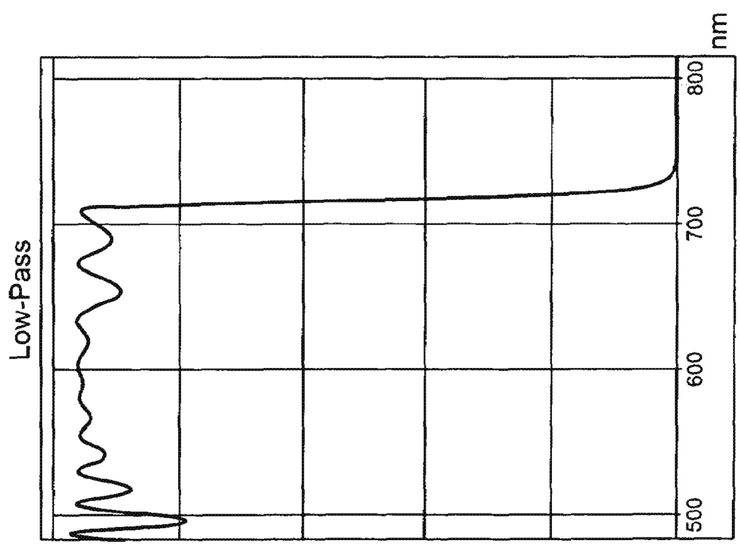
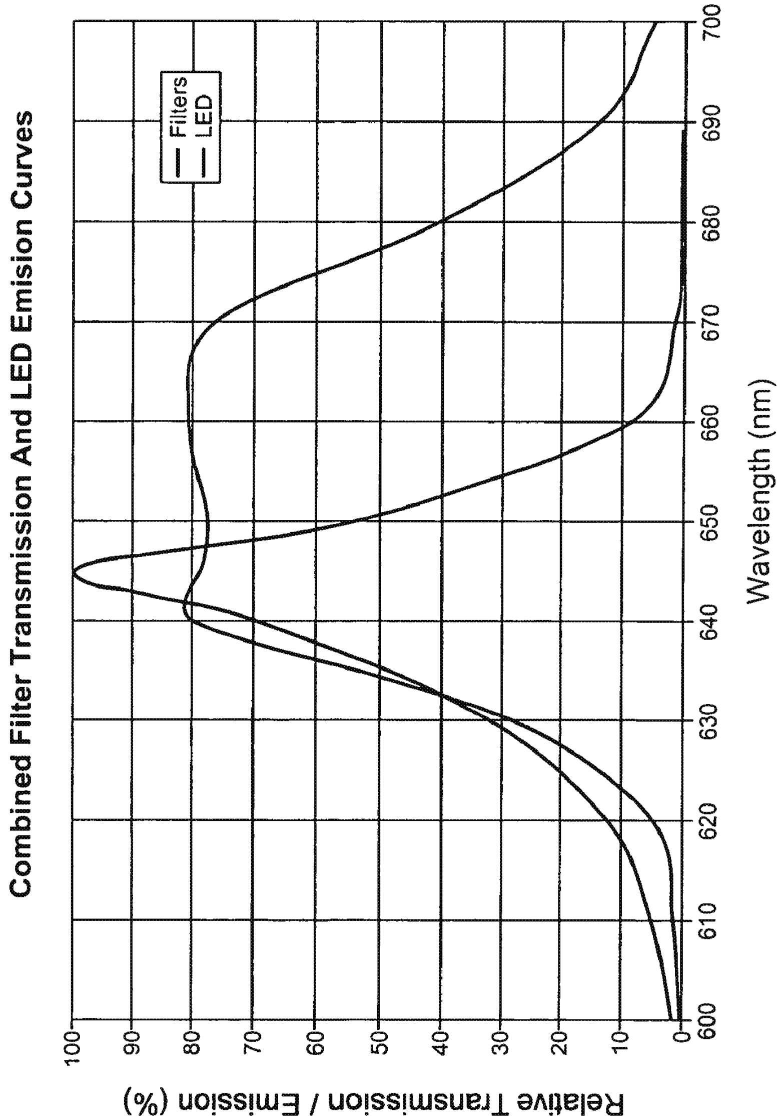


FIG. 6A2



($\Delta\lambda$) Bandwidth Of LED Emission Signal≈15 nmeters

FIG. 6A4

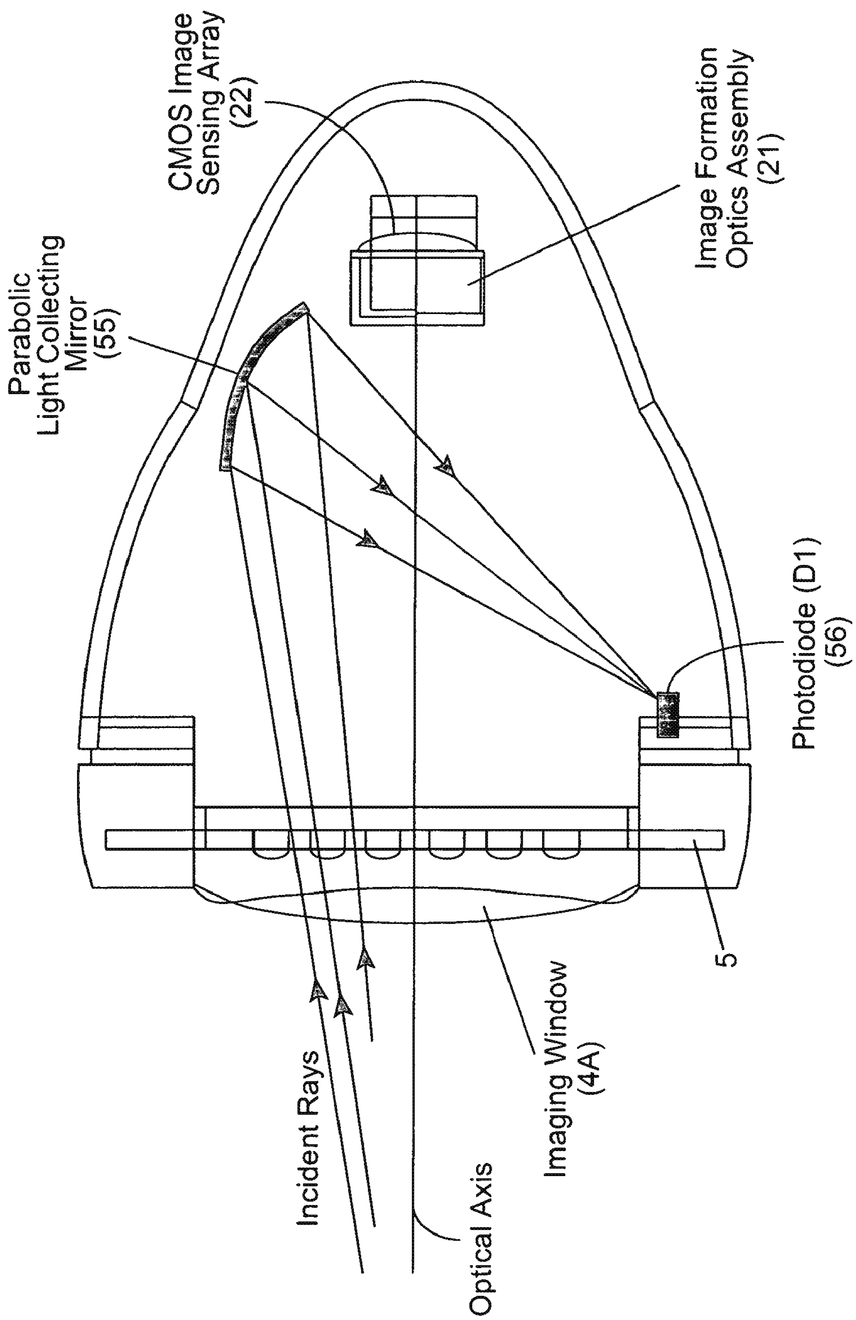

CMOS Image Sensing Array (22)
Image Formation Optics Assembly (21)
Parabolic Light Collecting Mirror (55)
Photodiode (D1) (56)
5
Imaging Window (4A)
Incident Rays
Optical Axis

FIG. 7A

FIG. 7A1

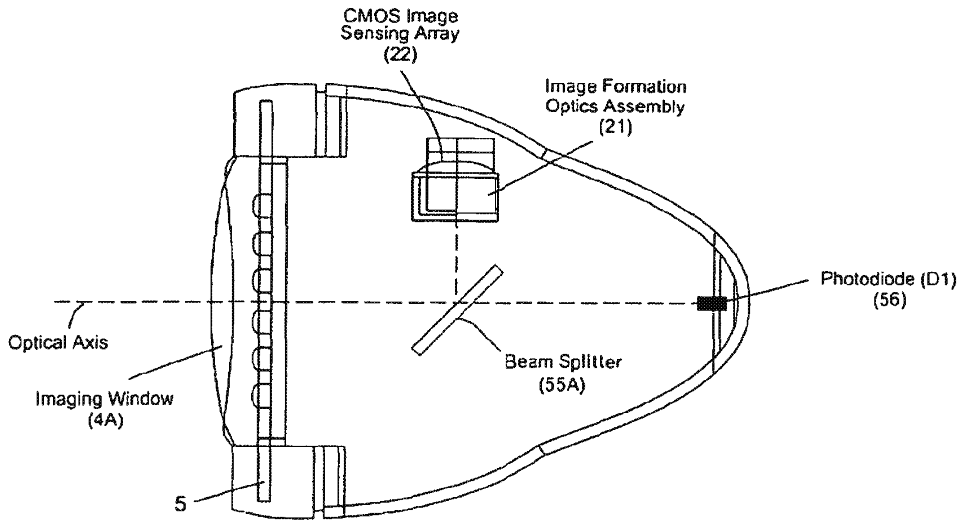
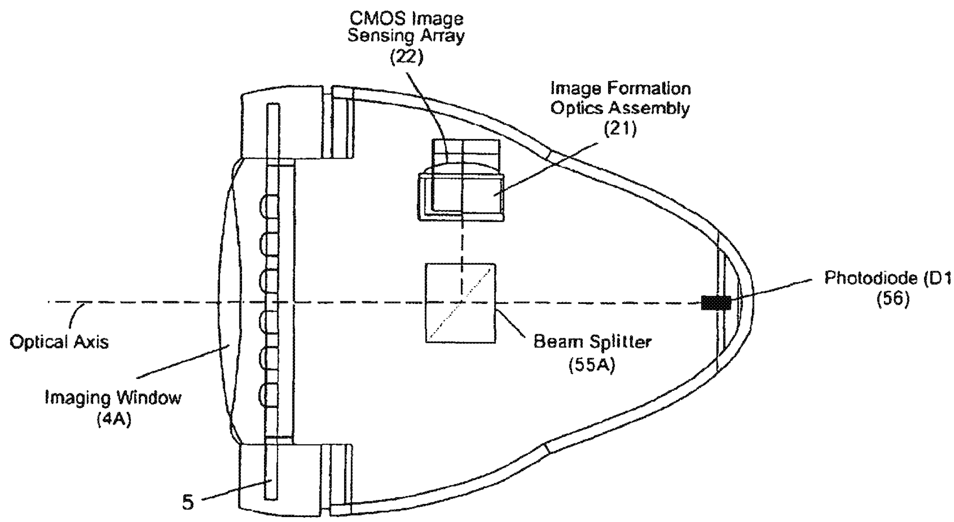
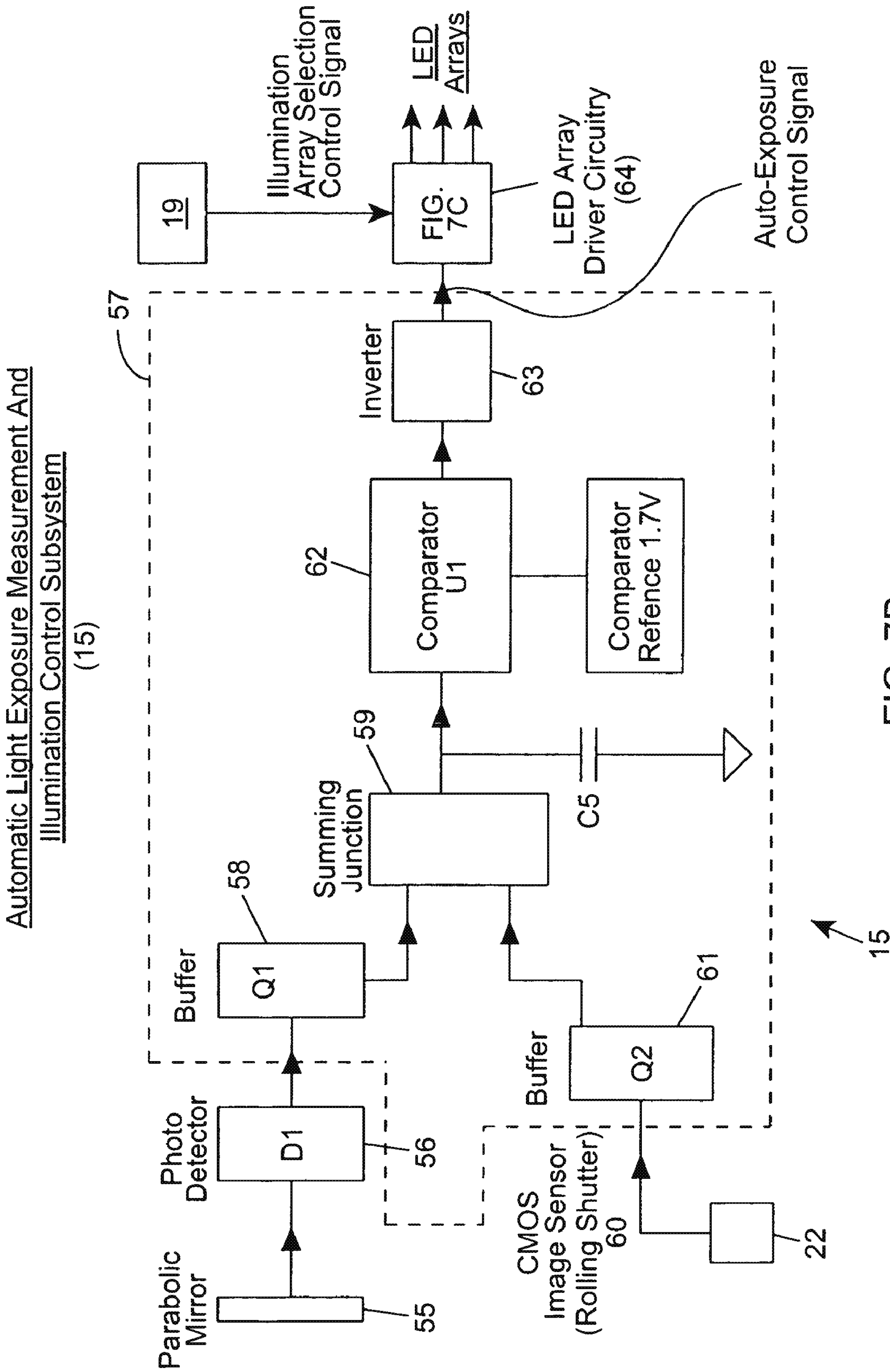


FIG. 7A2





Automatic Light Exposure Measurement And
Illumination Control Subsystem
(15)
57
Illumination
Array Selection
Control Signal
19
FIG.
7C
LED
Arrays
LED Array
Driver Circuitry
(64)
Auto-Exposure
Control Signal
Inverter
63
62
Comparator
U1
Comparator
Refence 1.7V
59
Summing
Junction
C5
58
Buffer
Q1
Photo
Detector
D1
56
Parabolic
Mirror
55
Buffer
Q2
61
CMOS
Image Sensor
(Rolling Shutter)
60
22
15

FIG. 7B

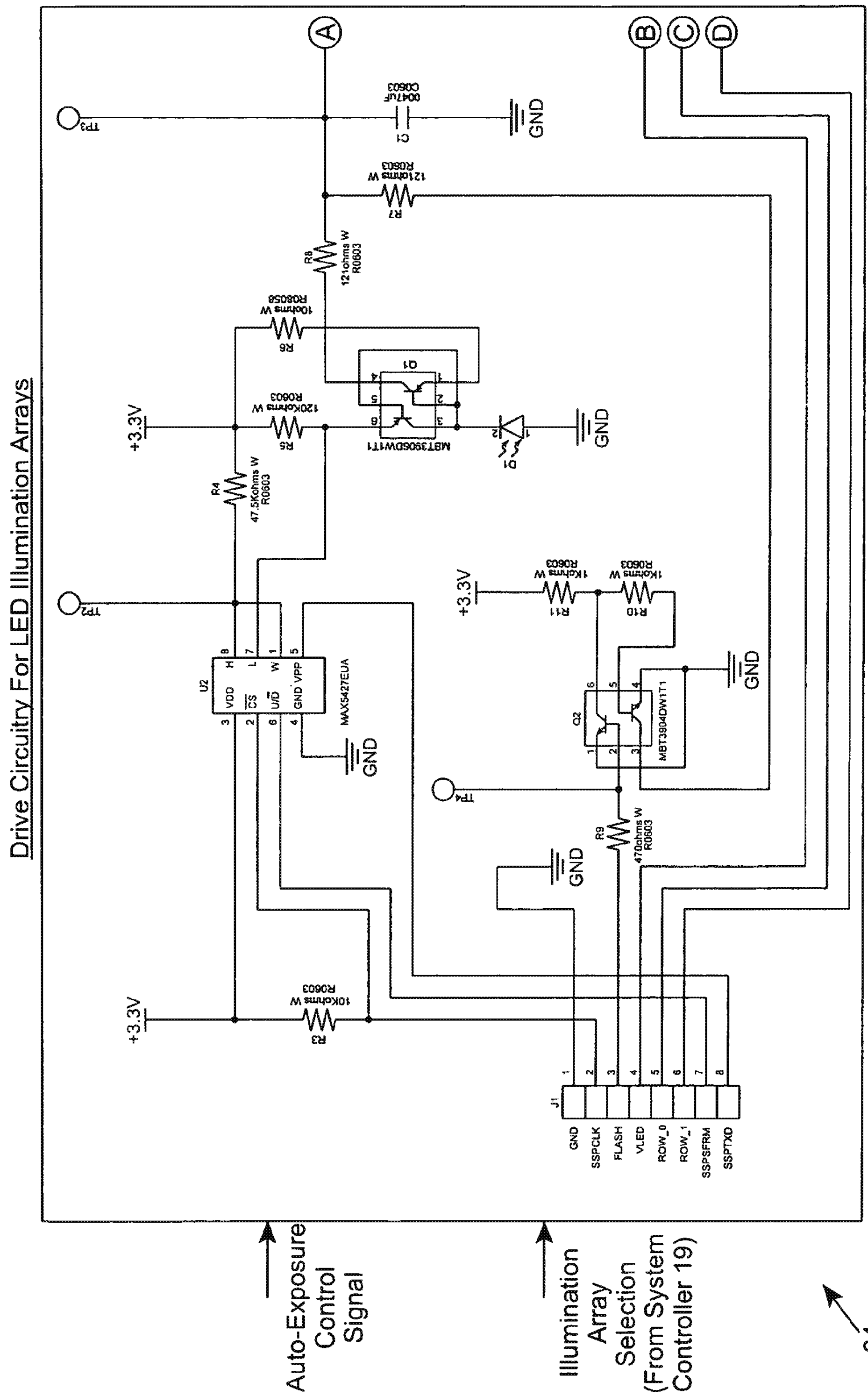

Drive Circuitry For LED Illumination Arrays
Auto-Exposure Control Signal
Illumination Array Selection (From System Controller 19)
64
MAX5427EUA
MBT3906DW1T1
MBT3904DW1T1
GND
SSPCLK
FLASH
VLED
ROW_0
ROW_1
SSPSFRM
SSPTXD

FIG. 7C1

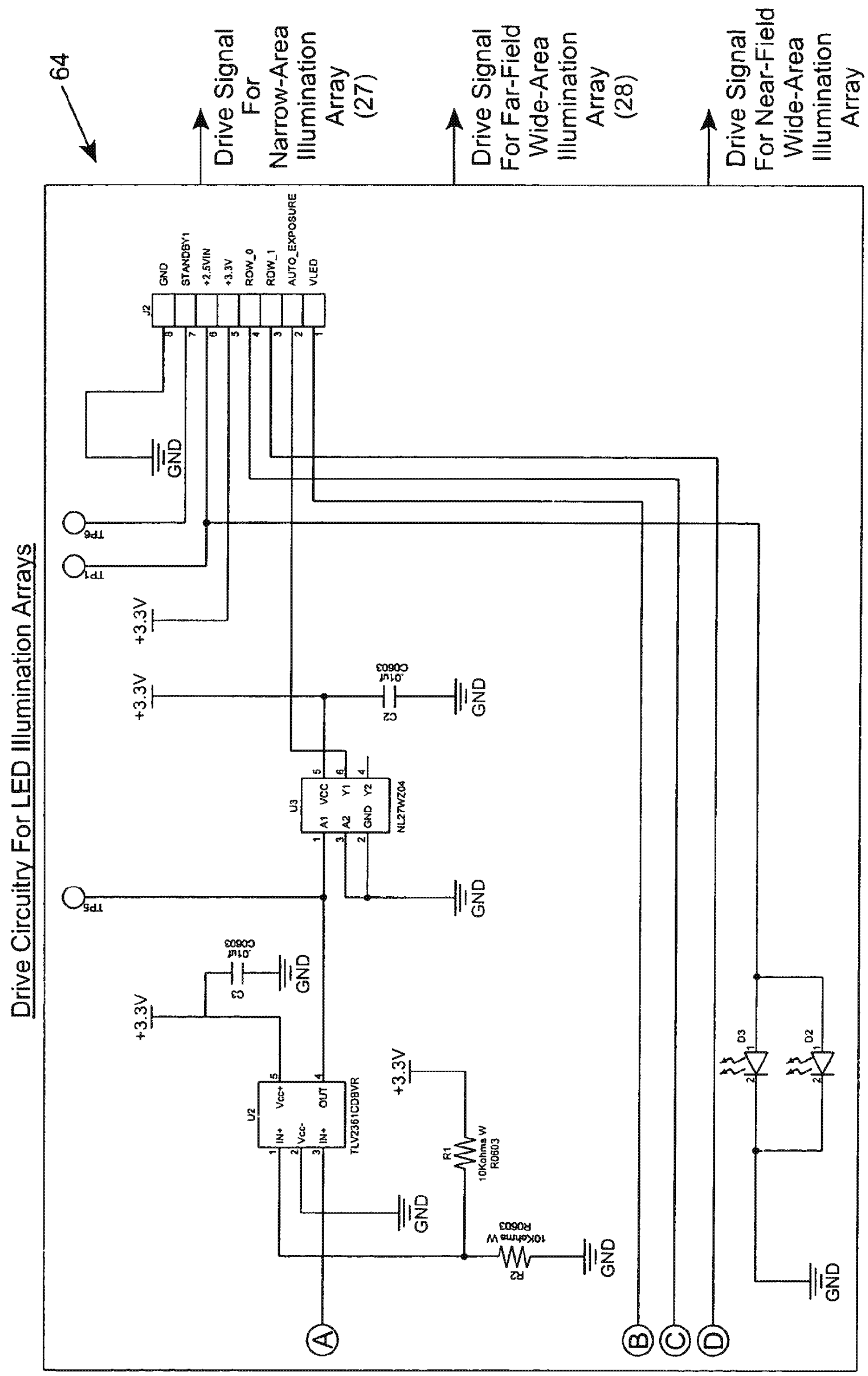

64
Drive Circuitry For LED Illumination Arrays
Drive Signal For Narrow-Area Illumination Array (27)
Drive Signal For Far-Field Wide-Area Illumination Array (28)
Drive Signal For Near-Field Wide-Area Illumination Array
J2
GND
STANDBY1
+2.5VIN
+3.3V
ROW_0
ROW_1
AUTO_EXPOSURE
VLED
TP6
TP1
TP5
+3.3V
GND
U3
NL27WZ04
U2
TLV2361CDBVR
C2
C3
.01uf
C0603
R1
R2
10Kohms W
R0603
D3
D2
A
B
C
D

FIG. 7C2

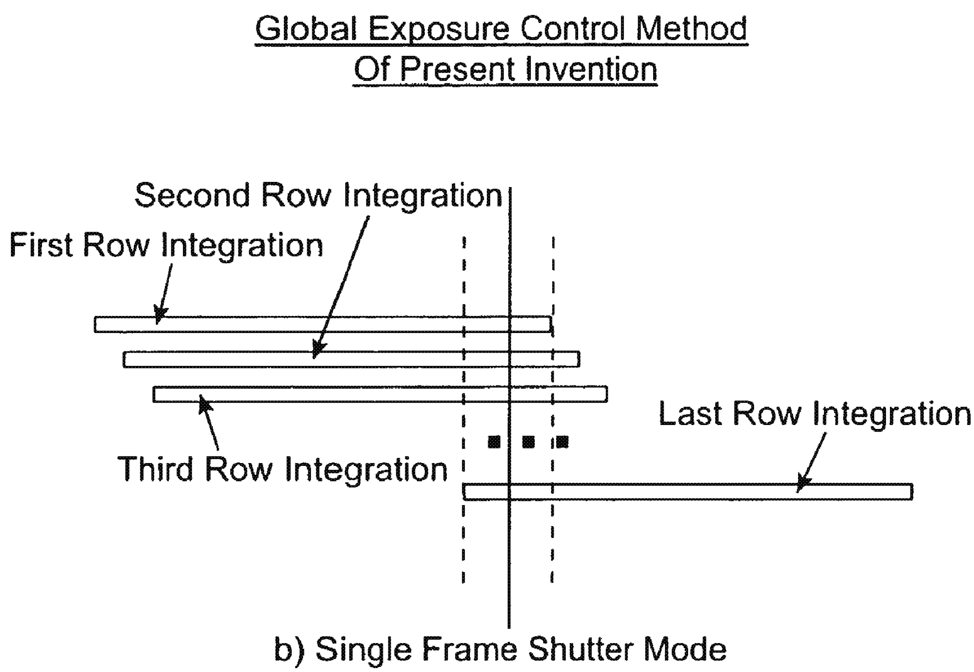

Global Exposure Control Method
Of Present Invention
Second Row Integration
First Row Integration
Last Row Integration
Third Row Integration
b) Single Frame Shutter Mode

FIG. 7D

METHOD OF GLOBAL EXPOSURE CONTROL
WITHIN A IMAGING-BASED BAR CODE SYMBOL READING SYSTEM

A — STEP A: SELECT THE SINGLE FRAME SHUTTER MODE OF OPERATION FOR THE CMOS IMAGING SENSING ARRAY PROVIDED WITHIN AN IMAGING-BASED BAR CODE SYMBOL READING SYSTEM EMPLOYING AN AUTOMATIC LIGHT EXPOSURE MEASUREMENT AND ILLUMINATION CONTROL SUBSYSTEM, A MULTI-MODE ILLUMINATION SUBSYSTEM, AND A SYSTEM CONTROL SUBSYSTEM INTEGRATED THEREWITH, AND IMAGE FORMATION OPTICS PROVIDING THE CMOS IMAGE SENSING ARRAY WITH A FIELD OF VIEW INTO A REGION OF SPACE WHERE OBJECTS TO BE IMAGED ARE PRESENTED.

↓

B — STEP B: USE THE AUTOMATIC LIGHT EXPOSURE MEASUREMENT AND ILLUMINATION CONTROL SUBSYSTEM TO CONTINUOUSLY COLLECT ILLUMINATION FROM A PORTION OF THE FIELD OF VIEW, DETECT THE INTENSITY OF THE COLLECTED ILLUMINATION, AND GENERATE AN ELECTRICAL ANALOG SIGNAL CORRESPONDING TO THE DETECTED INTENSITY, FOR PROCESSING.

↓

C — STEP C: ACTIVATE (E.G. BY WAY OF THE SYSTEM CONTROL SUBSYSTEM 19 OR DIRECTLY BY WAY OF TRIGGER SWITCH 2C) THE CMOS IMAGE SENSING ARRAY SO THAT ITS ROWS OF PIXELS BEGIN TO INTEGRATE PHOTONICALLY GENERATED ELECTRICAL CHARGE IN RESPONSE TO THE FORMATION OF AN IMAGE ONTO THE CMOS IMAGE SENSING ARRAY BY THE IMAGE FORMATION OPTICS OF THE SYSTEM.

↓

D — STEP D: WHEN ALL ROWS OF PIXELS IN THE IMAGE SENSING ARRAY ARE OPERATED IN A STATE OF INTEGRATION, AUTOMATICALLY GENERATE AN ELECTRONIC ROLLING SHUTTER (ERS) DIGITAL PULSE SIGNAL FROM THE CMOS IMAGE SENSING ARRAY AND PROVIDE THIS ERS PULSE SIGNAL TO THE AUTOMATIC LIGHT EXPOSURE MEASUREMENT AND ILLUMINATION CONTROL SUBSYSTEM SO AS TO ACTIVATE LIGHT EXPOSURE MEASUREMENT AND ILLUMINATION CONTROL OPERATIONS THEREWITHIN

↓

A

FIG. 7E1

A

E

STEP E: UPON ACTIVATION OF THE AUTOMATIC LIGHT EXPOSURE MEASUREMENT AND ILLUMINATION CONTROL SUBSYSTEM, PROCESS THE ELECTRICAL ANALOG SIGNAL BEING CONTINUOUSLY GENERATED THEREWITHIN, MEASURE THE LIGHT EXPOSURE WITHIN A PORTION OF SAID FIELD OF VIEW, AND GENERATE AN AUTO-EXPOSURE CONTROL SIGNAL FOR CONTROLLING THE GENERATION OF ILLUMINATION FROM AT LEAST ONE LED-BASED ILLUMINATION ARRAY IN THE MULTI-MODE ILLUMINATION SUBSYSTEM THAT IS SELECTED BY AN ILLUMINATION ARRAY SELECTION CONTROL SIGNAL PRODUCED BY THE SYSTEM CONTROL SUBSYSTEM

F

STEP: F: USE THE AUTO-EXPOSURE CONTROL SIGNAL AND THE ILLUMINATION ARRAY SELECTION CONTROL SIGNAL TO DRIVE THE SELECTED LED-BASED ILLUMINATION ARRAY AND GENERATE ILLUMINATION THEREFROM INTO THE FIELD OF VIEW OF THE CMOS IMAGE SENSING ARRAY, PRECISELY WHEN ALL ROWS OF PIXELS IN THE CMOS IMAGE SENSING ARRAY ARE IN A STATE OF INTEGRATION, THEREBY ENSURING THAT ALL ROWS OF PIXELS IN THE CMOS IMAGE SENSING ARRAY HAVE A COMMON INTEGRATION TIME.

FIG. 7E2

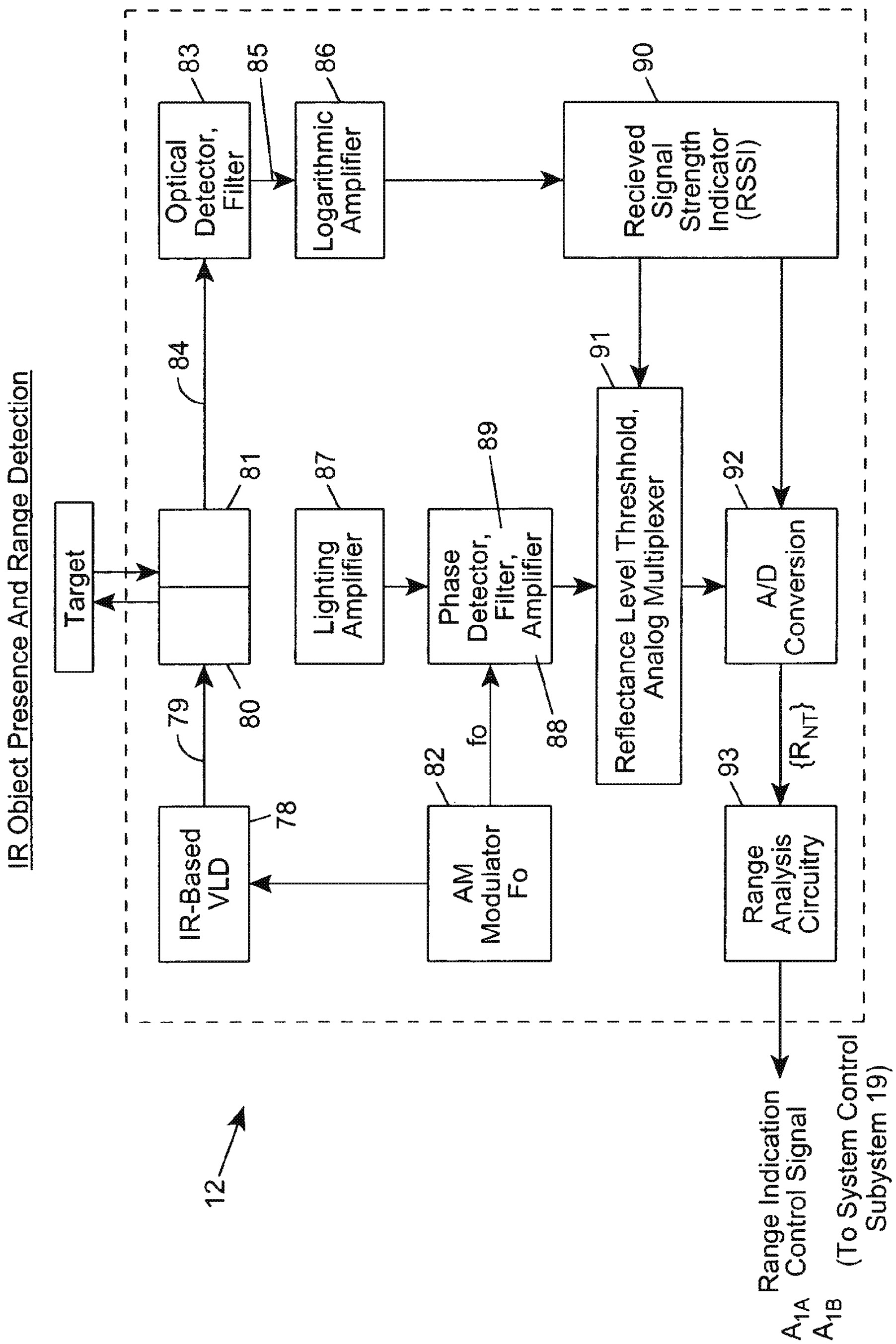
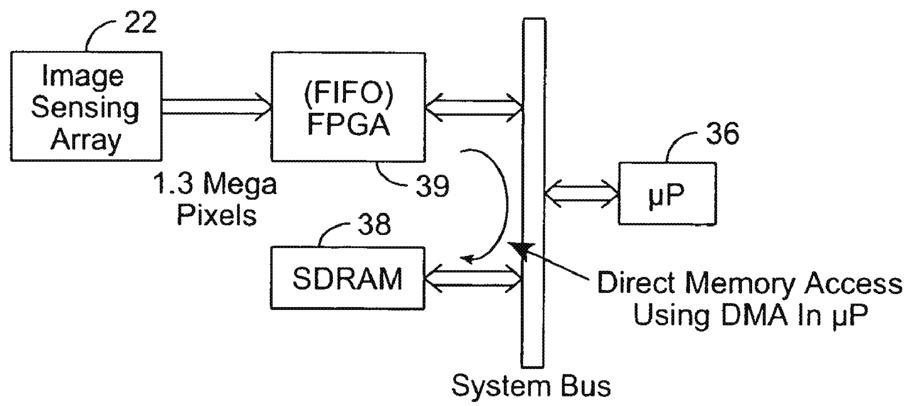
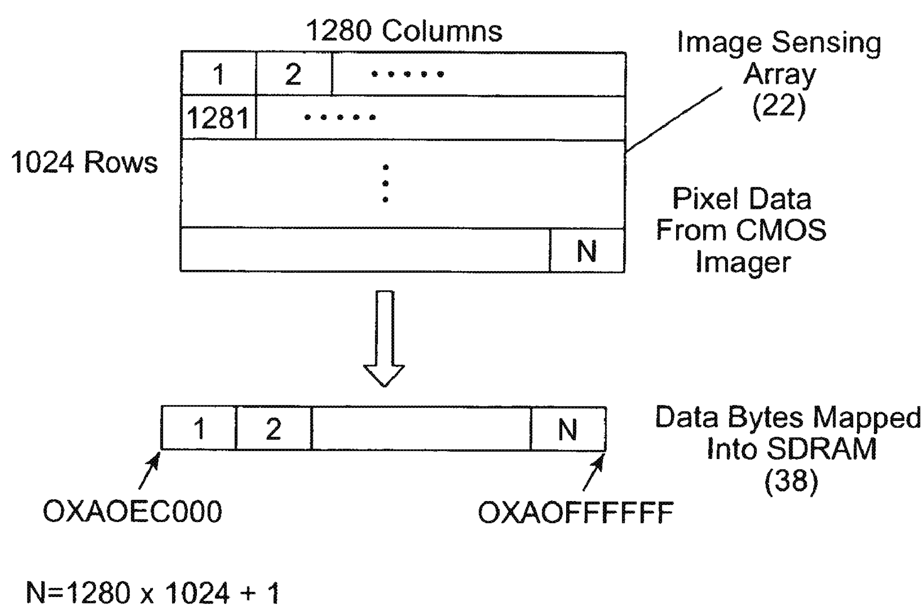

IR Object Presence And Range Detection
Target
Optical Detector, Filter
Logarithmic Amplifier
Recieved Signal Strength Indicator (RSSI)
Lighting Amplifier
Phase Detector, Filter, Amplifier
Reflectance Level Threshhold, Analog Multiplexer
A/D Conversion
IR-Based VLD
AM Modulator Fo
Range Analysis Circuitry
fo
$\{R_{NT}\}$
Range Indication Control Signal
A_{1A}
A_{1B}
(To System Control Subystem 19)
12
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93

FIG. 8



22
Image Sensing Array
(FIFO) FPGA
1.3 Mega Pixels
39
36
μP
38
SDRAM
Direct Memory Access Using DMA In μP
System Bus

FIG. 9



1280 Columns
1
2
1281
1024 Rows
N
Image Sensing Array (22)
Pixel Data From CMOS Imager
1
2
N
Data Bytes Mapped Into SDRAM (38)
OXAOEC000
OXAOFFFFFF
N=1280 x 1024 + 1

FIG. 10

U.S. Patent Jan. 6, 2009 Sheet 55 of 119 US 7,472,831 B2

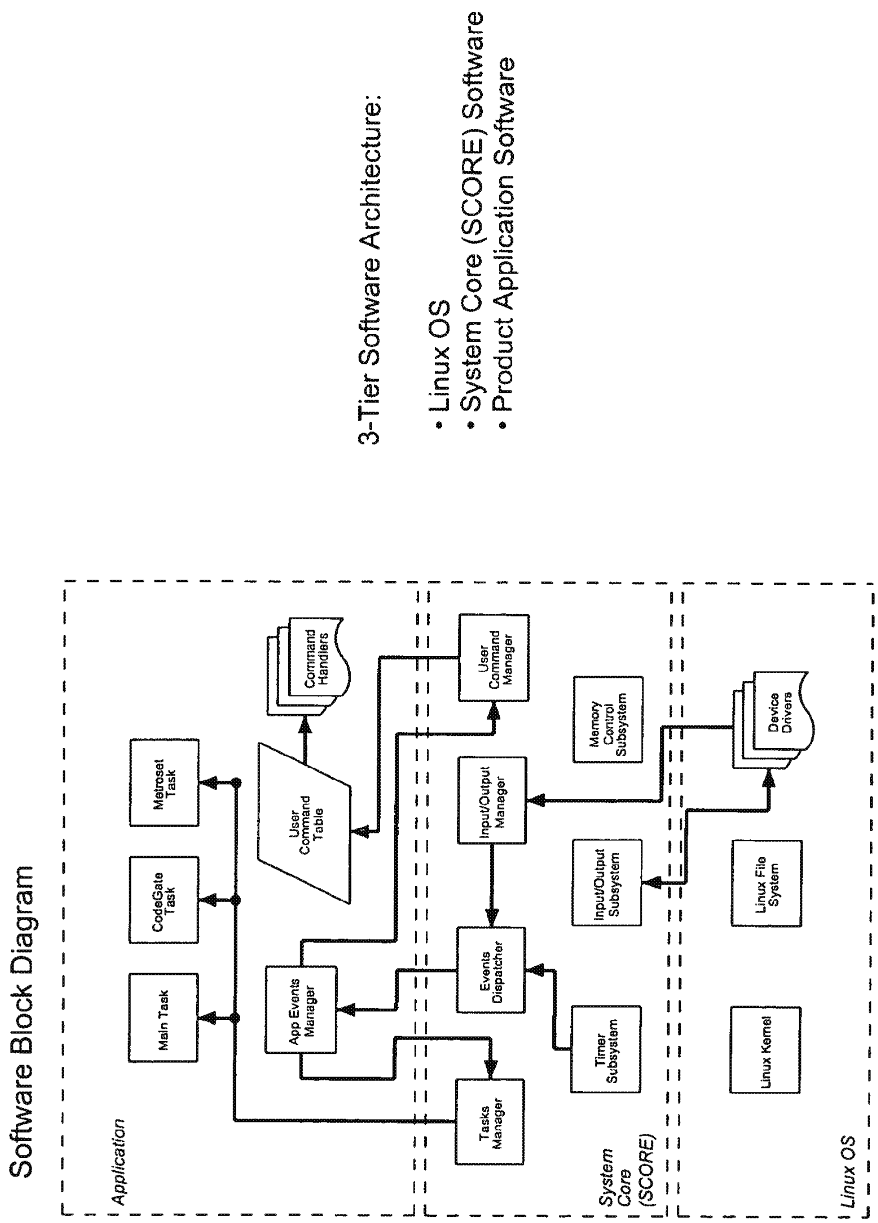


FIG. 11

Events Dispatcher

Provides a means of signaling and delivering events to the App Events Manager
(pointer to App Events Manager is provided at the SCORE initialization)

```
int
ScoreSignalEvent(int event_id,          /* Input: event id */
          void * p_par);          /* Input: pointer to the event's parameters */
```

App Events Manager is responsible for processing the event: It can start a new task, or stop currently running task, or do something or nothing and simply ignore the event.

FIG. 12A

Examples of System-Defined Events

SCORE_EVENT_POWER_UP

Signals the completion of the system start-up. No parameters.

SCORE_EVENT_TIMEOUT

Signals the timeout of the logical timer. Parameter: pointer to timer id.

SCORE_EVENT_UNEXPECTED_INPUT

Signals that the unexpected input data is available. Parameter: pointer to connection id.

SCORE_EVENT_TRIG_ON

Signals that the user pulled the trigger. No parameters.

SCORE_EVENT_TRIG_OFF

Signals that the user released the trigger. No parameters.

SCORE_EVENT_OBJECT_DETECT_ON

Signals that the object is positioned under the camera. No parameters.

SCORE_EVENT_OBJECT_DETECT_OFF

Signals that the object is removed from the field-of view of the camera. No parameters.

SCORE_EVENT_EXIT_TASK and **SCORE_EVENT_ABORT_TASK**

Signal the end of the task execution. Parameter: pointer to the UTID.

FIG. 12B

Tasks Manager

Provides a means of executing and stopping application specific tasks (threads)

```
typedef void *                              /* Return: pointer to the set of returned parameters */
(*TASK_FUNC)(void *params);                 /* Input:  set of input parameters */

int                                         /* Return: 0 if successful, otherwise error code */
ScoreStartTask(TASK_FUNC task_func,/* Input:  pointer to the task's main function */
       int task_id,                         /* Input:  id assigned to the task by application */
      void *task_params,                    /* Input:  parameters passed to the task's main function */
      int task_owner,                       /* Input:  connection id of the task's owner */
      int task_priority,                    /* Input:  task's priority (must be 0 for now) */
      size_t stacksize,                     /* Input:  size of the stack, or 0 for default size */
      size_t heapsize,                      /* Input:  size of the heap, or 0 for default size */
      UTID *p_utid);                        /* Output: unique task identifier */

BOOL       /* Return:  TRUE if it kills the task, or FALSE if the task was not found */
ScoreKillTask(UTID pthread_id)               /* Input:  unique task identifer */
```

FIG. 12C

Input / Output Manager

- High priority thread running in the background and monitoring activities of the external devices and user connections
- Signals appropriate events to the application when such activities are detected

FIG. 12D

Input / Output Subsystem

Provides a means of creating and deleting input/output connections...

```
int                                        /* Return: connection id if successful, otherwise (-1) */
ScoreIomngrCreateConnection(int conn_type,       /* Input:  connection type */
    int fd,                                /* Input:  file descriptor of a device or a socket */
    int conn_state,                        /* Input:  initial state of the connection, the value controlled by application */
    void *properties);                             /* Input:  pointer to the connection properties */

int                                        /* Return: connection id if successful, otherwise (-1) */
ScoreInitRS232(char *full_name,                    /* Input:  full name of the device, such as "/dev/ttyS0" */
            RS232_PROP *rs232_prop);               /* Input: RS232 parameters */
```

FIG. 12E1

Input / Output Subsystem

...and communicating with the outside world

```
int                                          /* Return: number of bytes received */
ScoreIomngrGetData(int connection_id,  /* Input:  connection id, or -1 for the task owner */
            char *input_buffer,        /* Input:  pointer to the input buffer */
            int min_len,          /* Input:  minimum number of bytes to receive */
            int max_len,          /* Input:  maximum number of bytes to receive */
            BOOL echo,            /* Input:  TRUE if data should be echoed back to device, otherwise FALSE */
            int timeout_ms);      /* Input:  If not 0, number of milliseconds to wait */

int                                          /* Return: 0 if successful, or (-1) in case of error */
ScoreIomngrSendData(int connection_id,         /* Input:  connection id */
            char *p_data,         /* Input:  pointer to the data buffer */
            int len);             /* Input:  number of bytes to send */

void
ScoreIomngrSendStream(int stream_type,         /* Input:  type of output stream */
            char *p_data,                      /* Input:  pointer to the data buffer */
            int len);                          /* Input:  number of bytes to send */
```

FIG. 12E2

Timer Subsystem

Provides a means of creating, deleting...

```
int                                /* Return: timer id if successful, otherwise (-1) */
ScoreCreateTimer(int flags);       /* Input:  optional SCORE_TIMER_CONTINUOUS */

void
ScoreDeleteTimer(int timer_id);          /* Input:  timer id, must be >= 0 */

int                                /* Return: 0 if successful, otherwise (-1) */
ScoreStartTimer(int timer_id,      /* Input:  timer id */
                int time_ms);      /* Input:  timer value, in ms */

int                                /* Return: 0 if successful, otherwise (-1) */
ScoreStopTimer(int timer_id);      /* Input:  timer id */
```

FIG. 12F1

Timer Subsystem

...and utilizing logical timers

```
BOOL                                   /* Return: TRUE if the timer timed out, otherwise FALSE */
ScoreTimerTimedOut(int timer_id);      /* Input:  timer id */

int                    /* Return: time (in ms) left before the timer times out, or (-1) in case of error */
ScoreGetTimeLeft(int timer_id);                  /* Input:  timer id */

int          /* Return: time (in ms) gone since the timer has been started (or restarted), or (-1) in case of error */
ScoreGetTime(int timer_id);            /* Input:  timer id */

BOOL                   /* Return: TRUE if timer is stopped, otherwise FALSE */
ScoreIsTimerStopped(int timer_id);     /* Input:  timer id */
```

FIG. 12F2

Memory Control Subsystem

Provides a thread-level dynamic memory management (the interfaces fully compatible with standard dynamic memory management functions)...

```
void *                          /* Return: pointer to the allocated memory if successful, otherwise NULL */
ScoreMalloc(size_t size);       /* Input:  size, in bytes, of the needed memory */

void
ScoreFree(void *mem);           /* Input:  pointer to the memory to be freed */
```

FIG. 12G1

Memory Control Subsystem

...as well as a means of buffering the data

```
int                                                        /* Return: 0 if successful */
ScoreCreateOutpMem(SCORE_OUTP_MEM *p_outp_mem);            /* Input: pointer to buffered memory structure */

void
ScoreDestroyOutpMem(SCORE_OUTP_MEM  *p_outp_mem);          /* Input: pointer to buffered memory structure */

int                                                 /* Return: 0 if successful */
ScoreWriteToOutpMem (SCORE_OUTP_MEM *p_outp_mem,           /* Input: pointer to buffered memory structure */
      void *p_data,                                 /* Input: pointer to the data to be buffered up for output */
      size_t len);                                  /* Input: size of the data, in bytes */

int                                                 /* Return: 0 if successful */
ScoreSendDataFromOutpMem(int connection_id,         /* Input: id of the connection to send the data to */
       SCORE_OUTP_MEM  *p_outp_mem);                /* Input: pointer to buffered memory structure */

int                                                 /* Return: 0 if successful */
ScoreSendStreamFromOutpMem(int stream_type,         /* Input:  type of output stream */
       SCORE_OUTP_MEM  *p_outp_mem);                /* Input: pointer to buffered memory structure */
```

FIG. 12G2

User Commands Manager

Provides a standard way of entering user commands and executing application modules responsible for handling them

(pointer to User Commands Table is provided at the SCORE initialization)

```
int
ScoreCmdManager(void *params);

rc = ScoreStartTask(ScoreCmdManager,      /* Input:  user command manager task */
        CMDMNGR_TASK_ID,                  /* Input:  id assigned to the commands manager */
        NULL,
        0,
        connection_id,                    /* Input:  connection id of the owner */
        0,                                /* Input:  priority */
        (64 * 1024),                      /* Input:  stack size */
        (512 * 1024),                     /* Input:  heap size */
        &cmdmngr_utid);                   /* Output: unique task identifier */
```

FIG. 12H

Device Drivers

- Trigger driver -- establishes software connection with the hardware trigger
- Image acquisition driver -- implements image acquisition functionality
- IR driver -- implements object detection functionality

FIG. 12I

Example of Flow of Events

- User points the scanner towards a barcode label
- Object is detected
- The IR device driver wakes up the Input/Output Manager

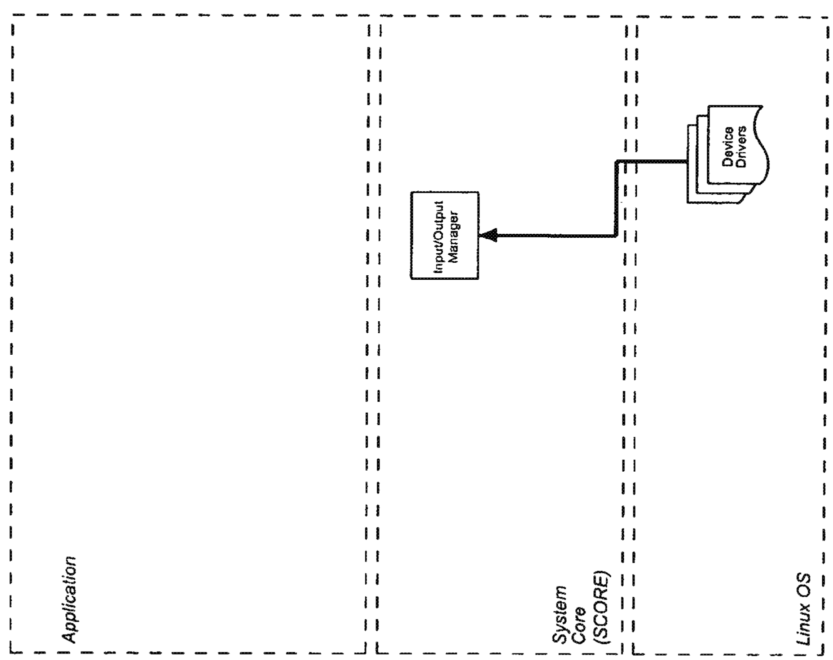
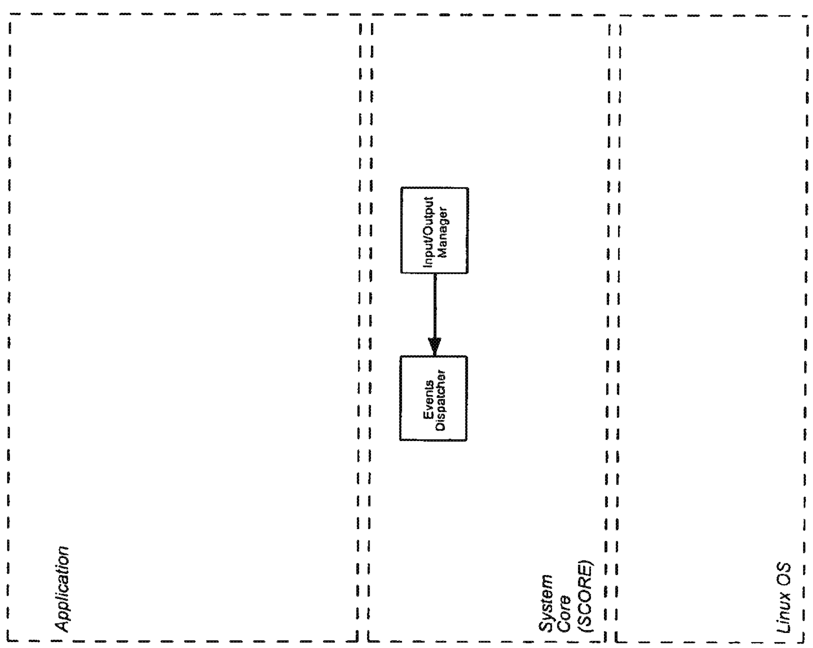


FIG. 13A

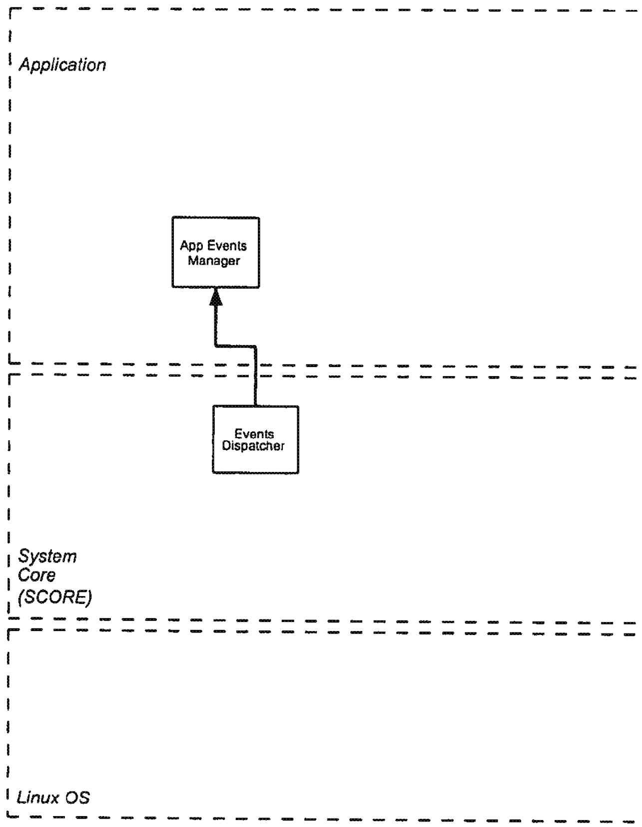
Example of Flow of Events



- The Input/Output Manager posts the SCORE_OBJECT_DETECT_ON event

FIG. 13B

Example of Flow of Events



• The Events Dispatcher passes the SCORE_OBJECT_DETECT_ON event to the application

FIG. 13C

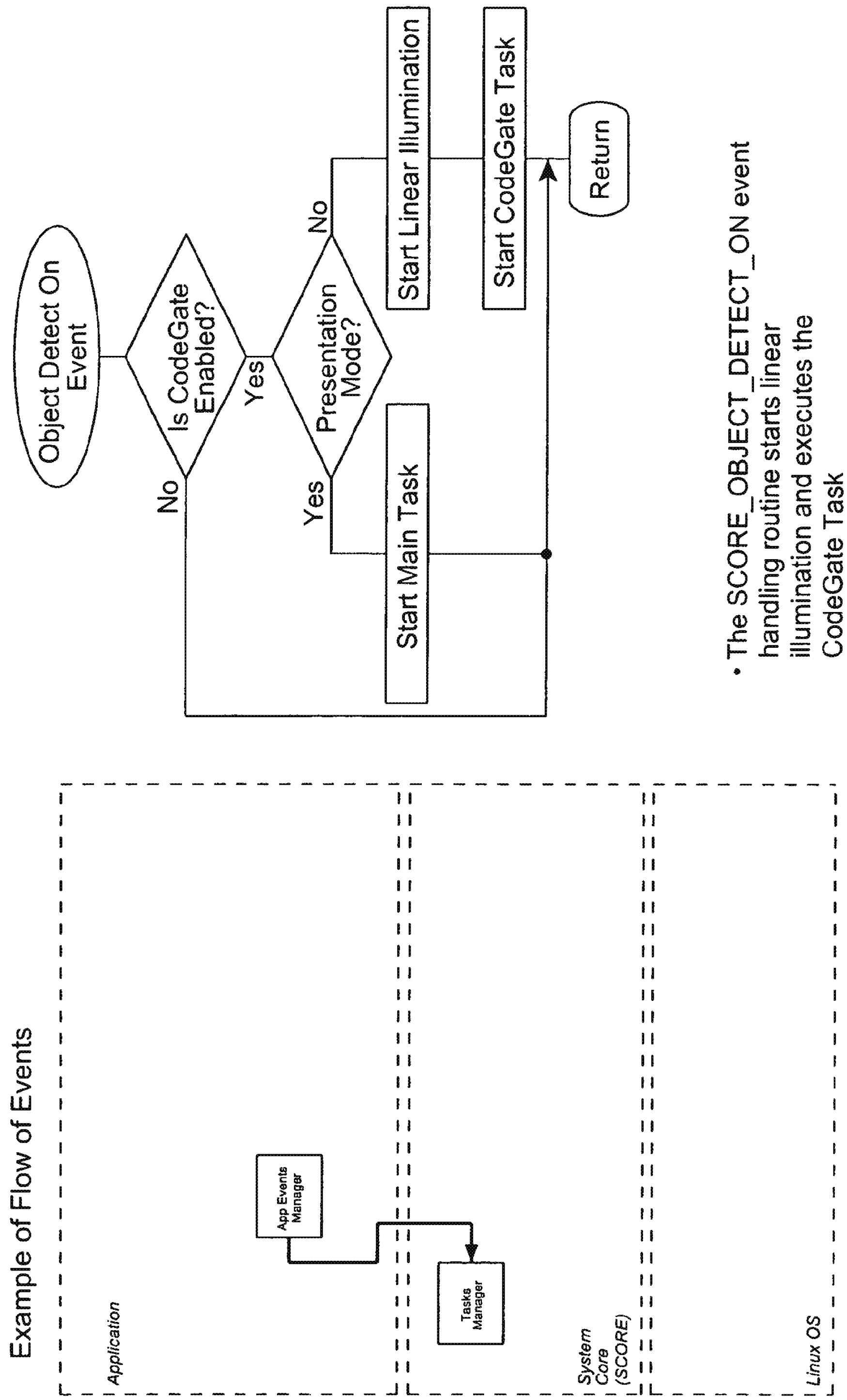

Example of Flow of Events
Application
App Events Manager
Tasks Manager
System Core (SCORE)
Linux OS
Object Detect On Event
Is CodeGate Enabled?
No
Yes
Presentation Mode?
Yes
No
Start Main Task
Start Linear Illumination
Start CodeGate Task
Return
• The SCORE_OBJECT_DETECT_ON event handling routine starts linear illumination and executes the CodeGate Task

FIG. 13D

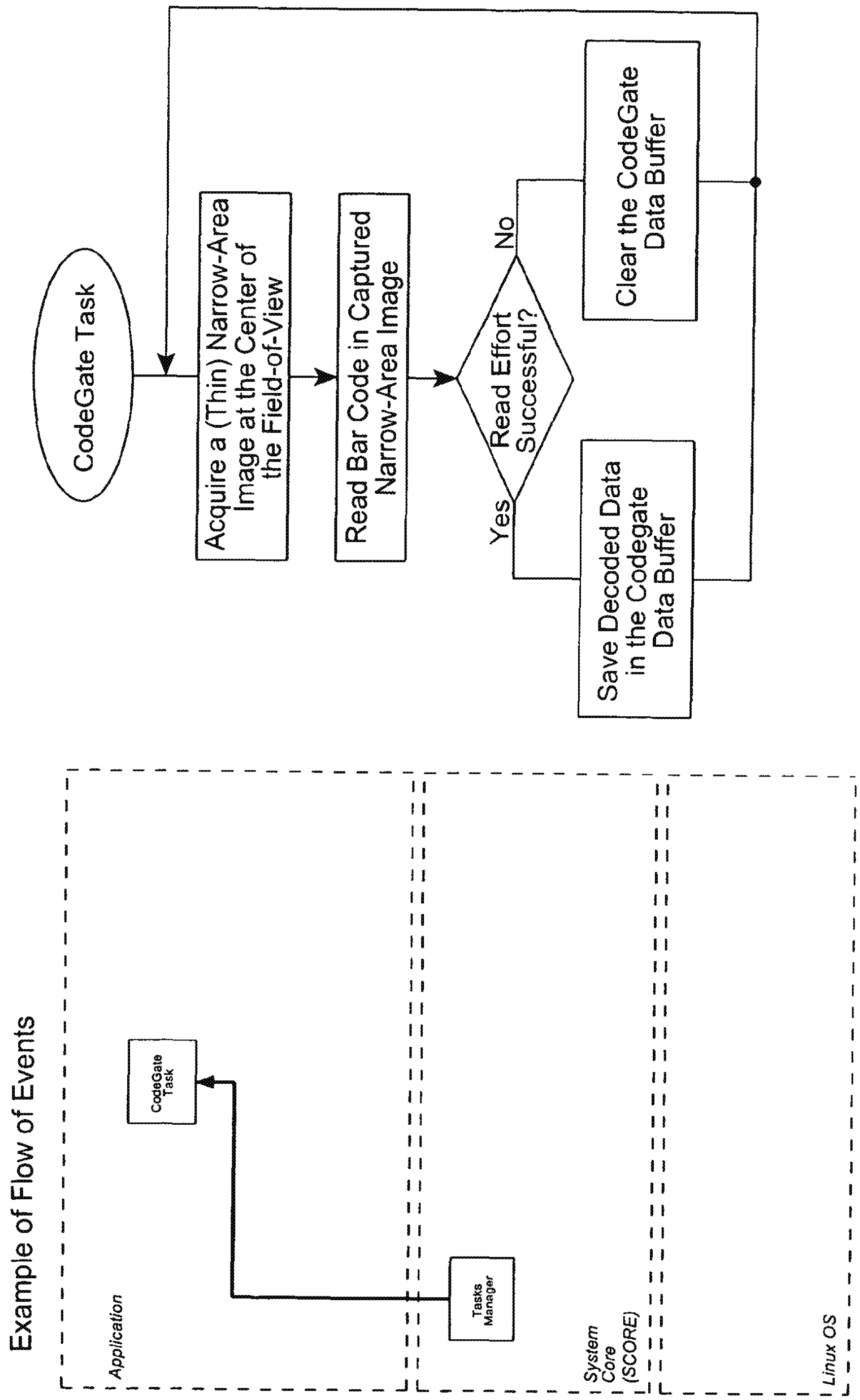

CodeGate Task
Acquire a (Thin) Narrow-Area Image at the Center of the Field-of-View
Read Bar Code in Captured Narrow-Area Image
Read Effort Successful?
No
Yes
Clear the CodeGate Data Buffer
Save Decoded Data in the Codegate Data Buffer
Example of Flow of Events
Application
CodeGate Task
Tasks Manager
System Core (SCORE)
Linux OS

FIG. 13E

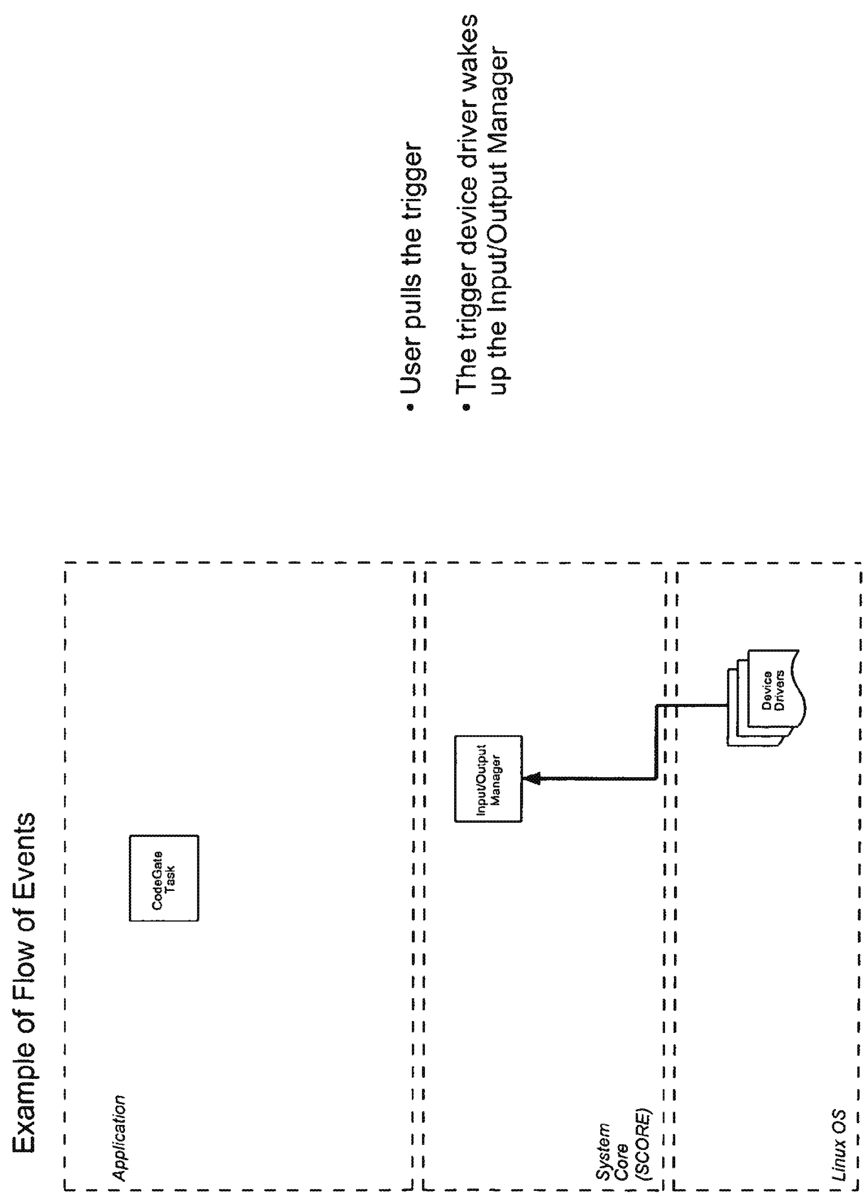
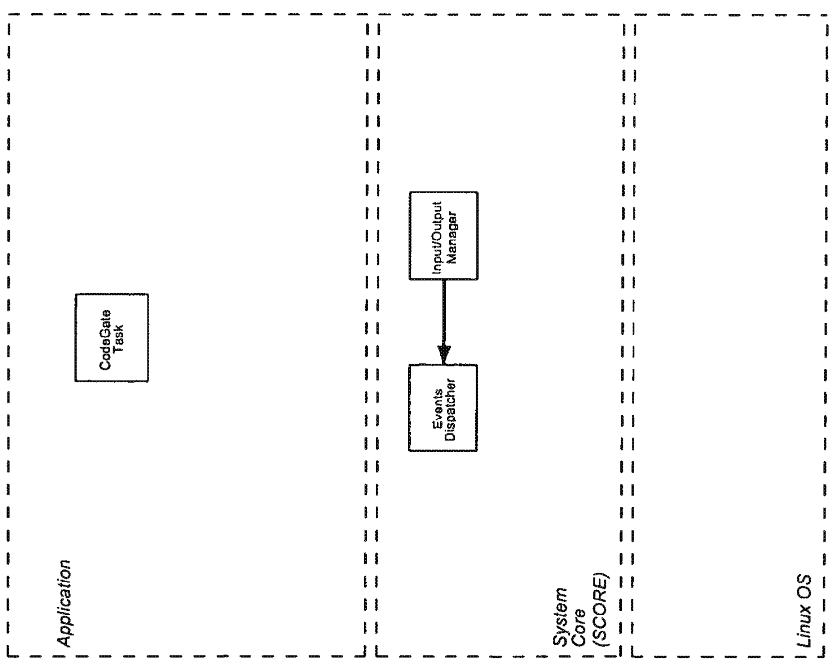

Example of Flow of Events
• User pulls the trigger
• The trigger device driver wakes up the Input/Output Manager
Application
CodeGate Task
System Core (SCORE)
Input/Output Manager
Linux OS
Device Drivers

FIG. 13F

Example of Flow of Events



- The Input/Output Manager posts the SCORE_TRIGGER_ON event

FIG. 13G

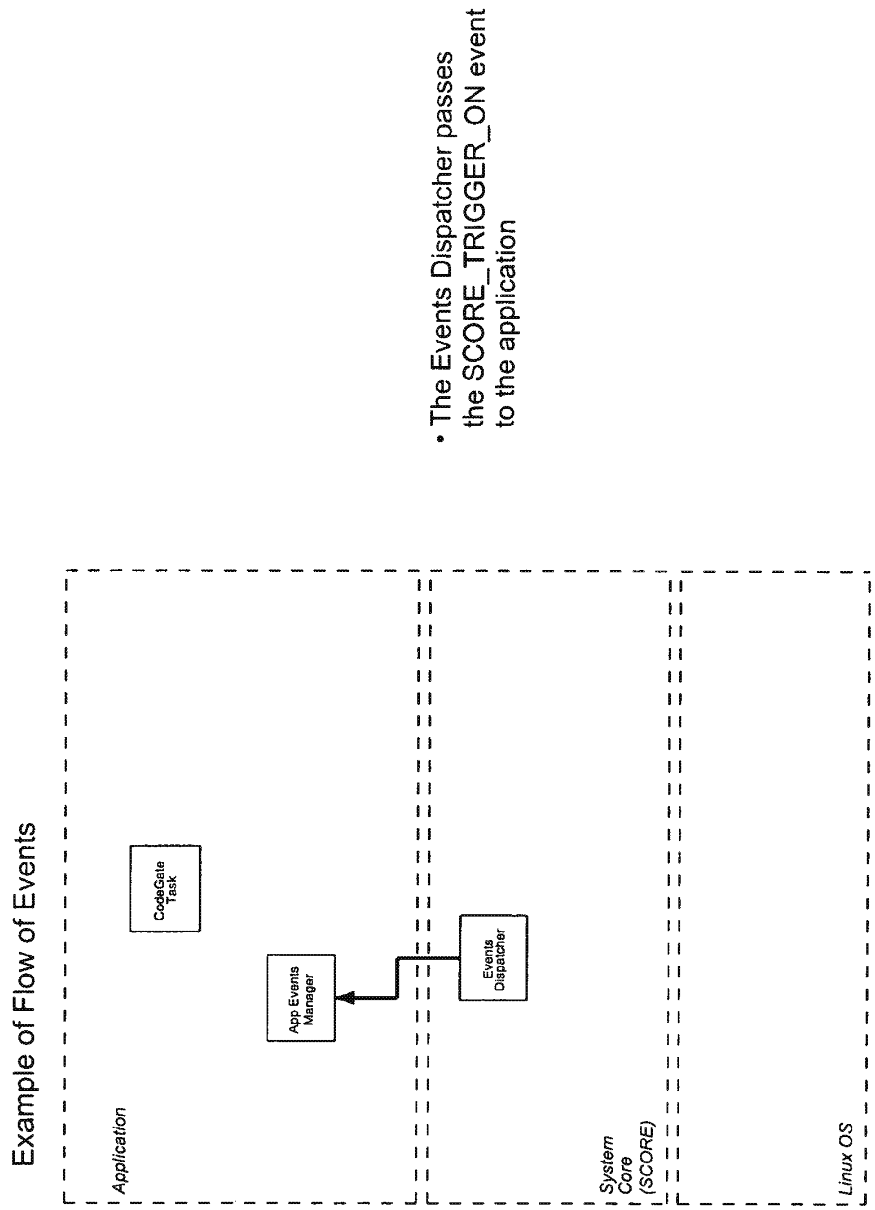

Example of Flow of Events
Application
CodeGate Task
App Events Manager
Events Dispatcher
System Core (SCORE)
Linux OS
• The Events Dispatcher passes the SCORE_TRIGGER_ON event to the application

FIG. 13H

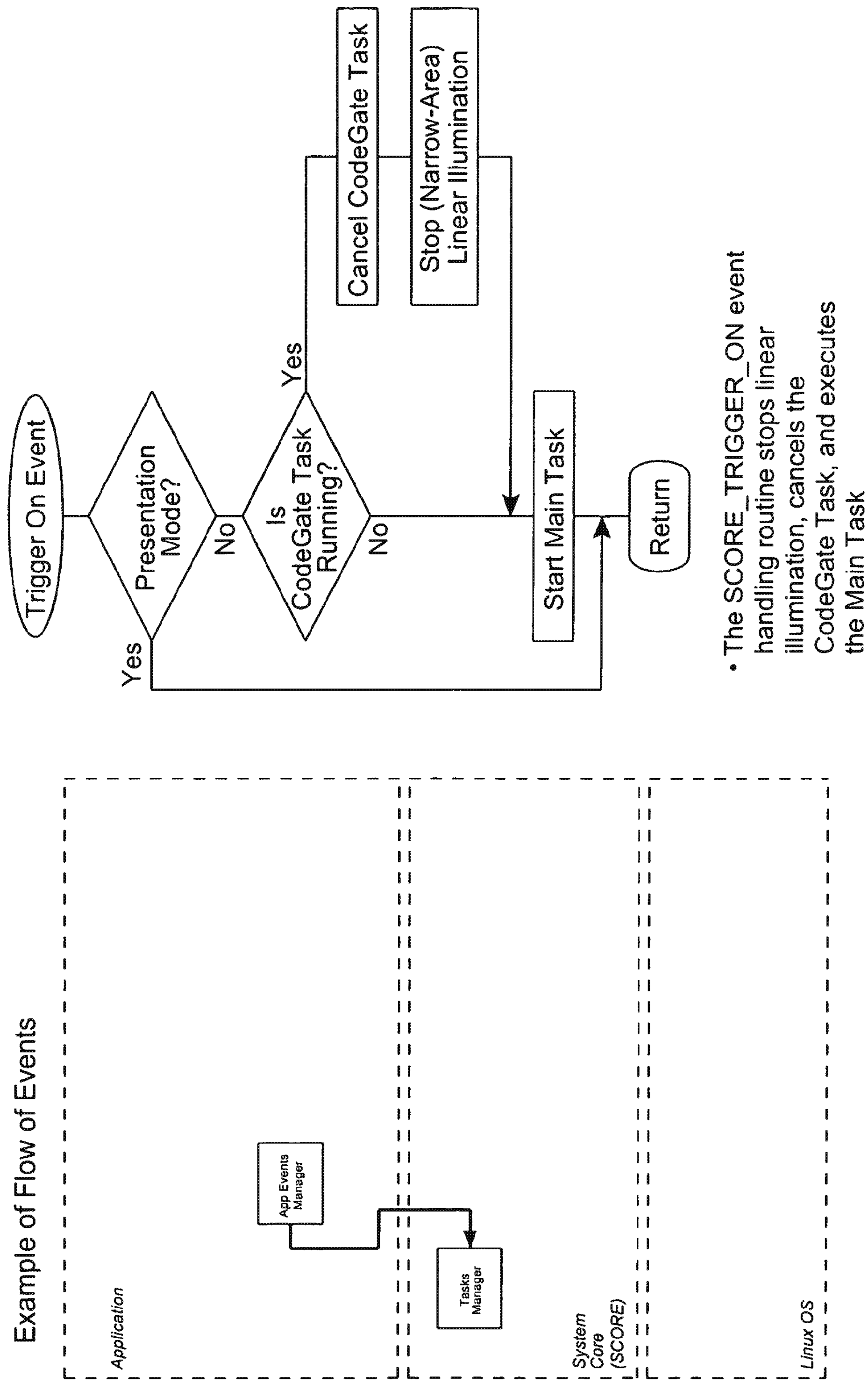

Trigger On Event
Presentation Mode?
Yes
No
Is CodeGate Task Running?
Yes
No
Cancel CodeGate Task
Stop (Narrow-Area) Linear Illumination
Start Main Task
Return
• The SCORE_TRIGGER_ON event handling routine stops linear illumination, cancels the CodeGate Task, and executes the Main Task
Example of Flow of Events
Application
App Events Manager
Tasks Manager
System Core (SCORE)
Linux OS

FIG. 13I

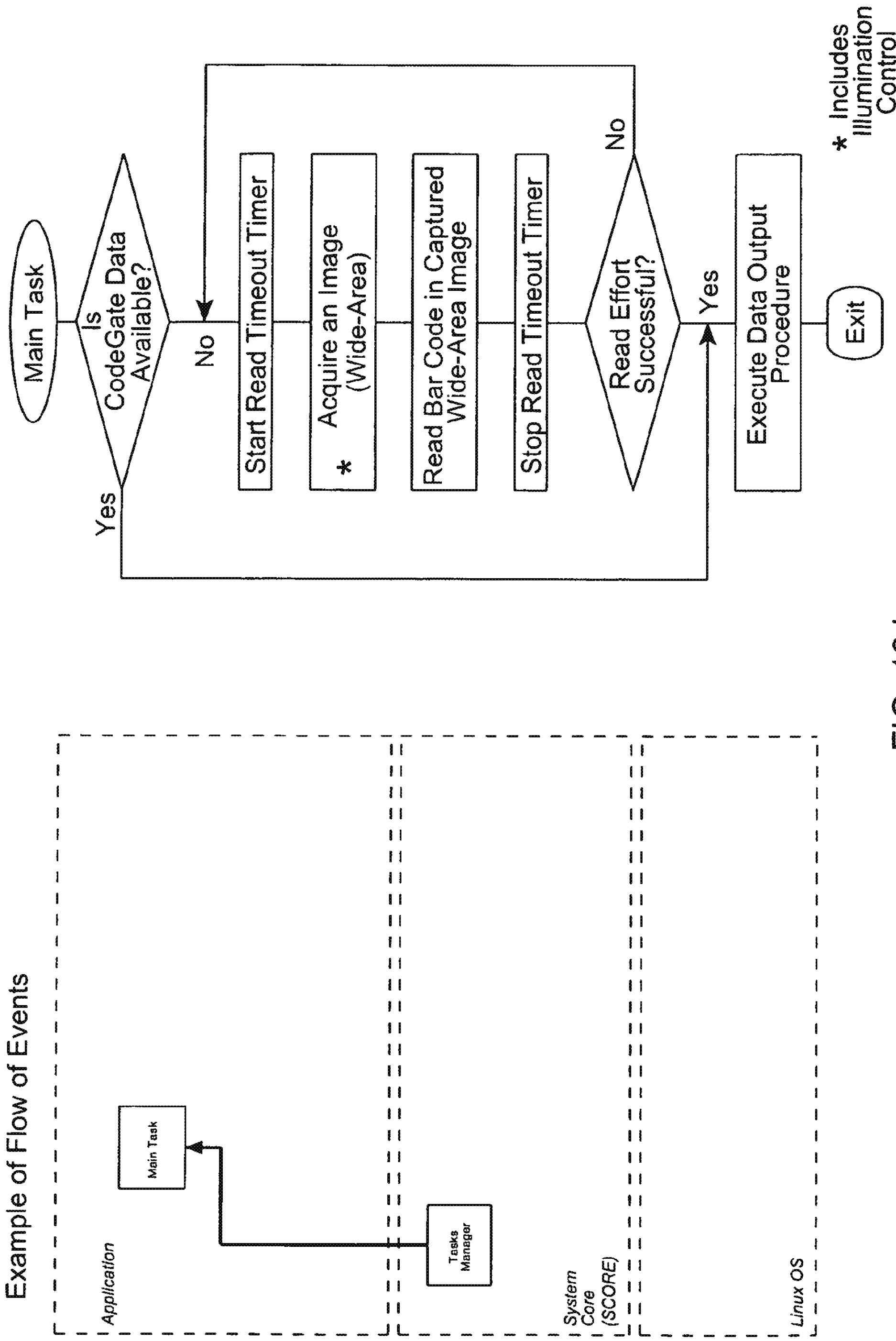

Main Task
Is CodeGate Data Available?
Yes
No
Start Read Timeout Timer
Acquire an Image (Wide-Area)
*
Read Bar Code in Captured Wide-Area Image
Stop Read Timeout Timer
Read Effort Successful?
No
Yes
Execute Data Output Procedure
Exit
* Includes Illumination Control
Example of Flow of Events
Application
Main Task
Tasks Manager
System Core (SCORE)
Linux OS

FIG. 13J

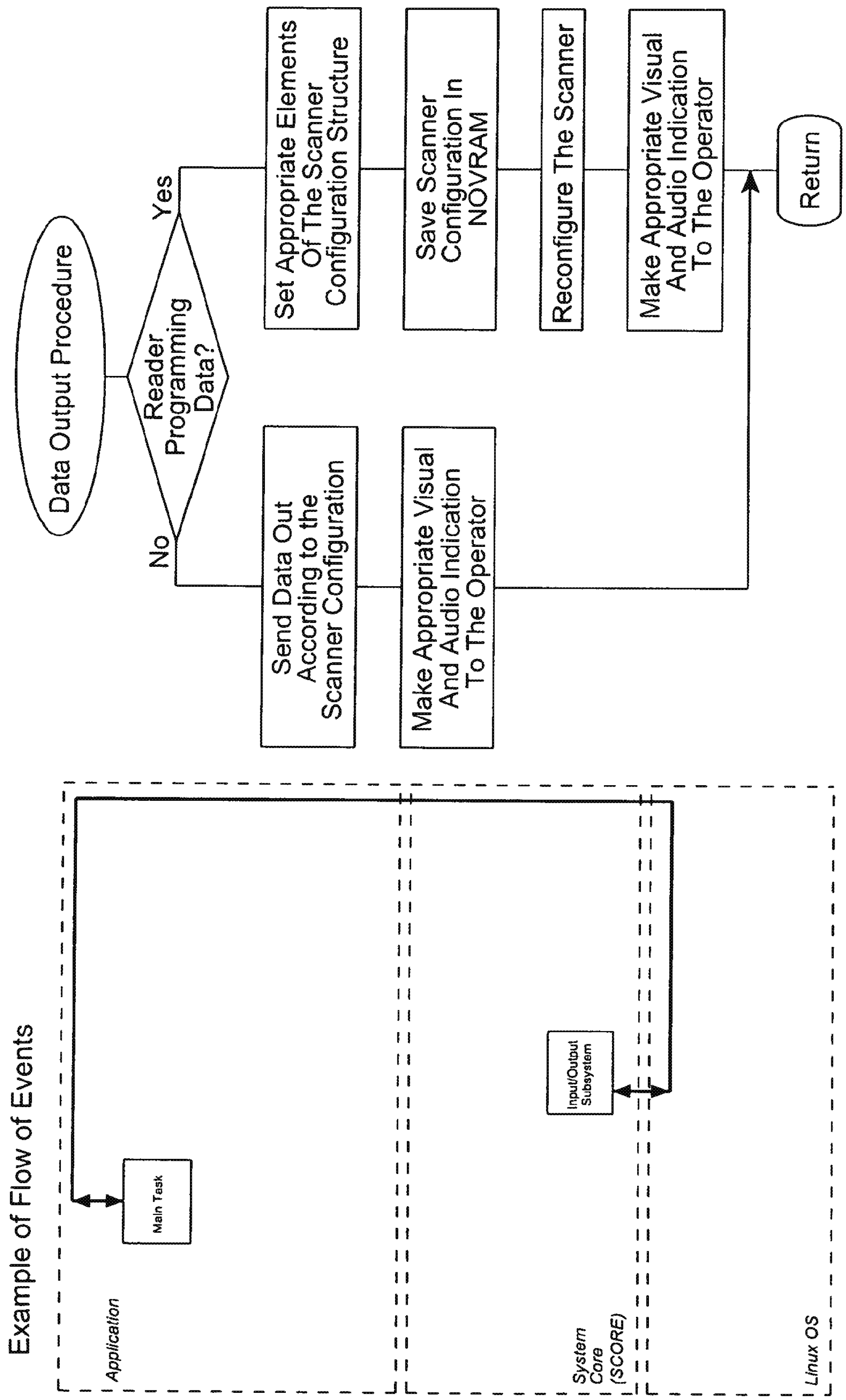

Data Output Procedure
Reader Programming Data?
Yes
No
Set Appropriate Elements Of The Scanner Configuration Structure
Save Scanner Configuration In NOVRAM
Reconfigure The Scanner
Make Appropriate Visual And Audio Indication To The Operator
Return
Send Data Out According to the Scanner Configuration
Make Appropriate Visual And Audio Indication To The Operator
Example of Flow of Events
Application
Main Task
System Core (SCORE)
Input/Output Subsystem
Linux OS

FIG. 13K

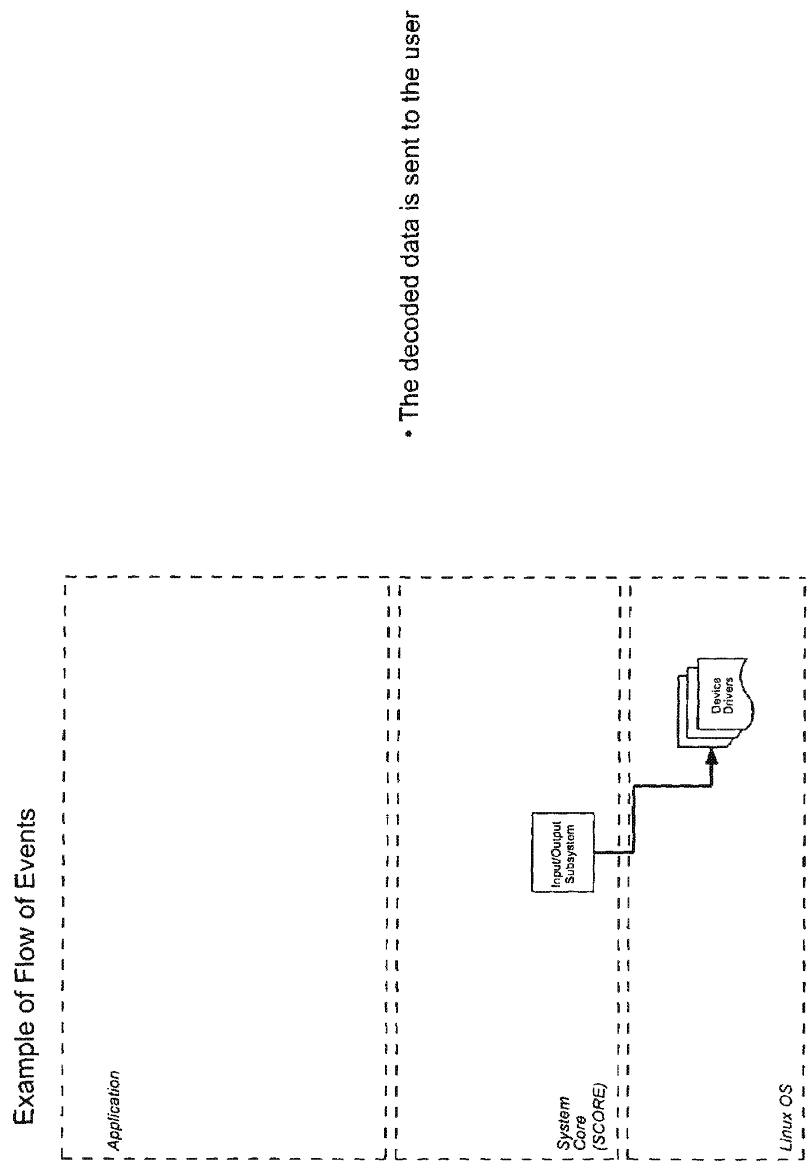

Example of Flow of Events
• The decoded data is sent to the user
Application
System Core (SCORE)
Input/Output Subsystem
Linux OS
Device Drivers

FIG. 13L

METHOD OF ILLUMINATING OBJECTS WITHOUT SPECULAR REFLECTION

STEP A: USE THE AUTOMATIC LIGHT EXPOSURE MEASUREMENT AND CONTROL SUBSYSTEM TO MEASURE THE LIGHT LEVEL TO WHICH THE CMOS IMAGE SENSING ARRAY IS EXPOSED.

STEP B: USE THE AUTOMATIC IR-BASED OBJECT PRESENCE AND RANGE DETECTION SUBSYSTEM TO MEASURE THE PRESENCE AND RANGE OF THE OBJECT IN EITHER THE NEAR OR FAR FIELD PORTION OF THE FIELD OF VIEW (FOV) OF THE SYSTEM.

STEP C: USE THE DETECTED RANGE AND THE MEASURED LIGHT EXPOSURE LEVEL TO DRIVE BOTH THE UPPER AND LOWER LED SUBARRAYS ASSOCIATED WITH EITHER THE NEAR OR FAR FIELD WIDE AREA ILLUMINATION ARRAY.

STEP D: CAPTURE A WIDE-AREA IMAGE AT THE CMOS IMAGE SENSING ARRAY USING THE ILLUMINATION FIELD PRODUCED DURING STEP C.

STEP E: RAPIDLY PROCESS THE CAPTURED WIDE-AREA IMAGE DURING STEP D TO DETECT THE OCCURANCE OF HIGH SPATIAL-INTENSITY LEVELS IN THE CAPTURED WIDE-AREA IMAGE, INDICATIVE OF A SPECULAR REFLECTION CONDITION.

STEP F:

IF A SPECULAR REFLECTION CONDITION IS DETECTED IN THE PROCESSED WIDE-AREA IMAGE, THEN DRIVE ONLY THE UPPER LED SUBARRAY ASSOCIATED WITH EITHER THE NEAR FIELD OR FAR FIELD WIDE AREA ILLUMINATION ARRAY.

IF A SPECULAR REFLECTION CONDITION IS NOT DETECTED IN THE PROCESSED WIDE-AREA IMAGE, THEN USE THE DETECTED RANGE AND THE MEASURED LIGHT EXPOSURE LEVEL TO DRIVE BOTH THE UPPER AND LOWER LED SUBARRAYS ASSOCIATED WITH EITHER THE NEAR FIELD OR FAR FIELD WIDE AREA ILLUMINATION ARRAY.

FIG. 13M1

STEP G: CAPTURE A WIDE-AREA IMAGE AT THE CMOS IMAGE SENSING ARRAY USING THE ILLUMINATION FIELD PRODUCED DURING STEP F.

STEP H: RAPIDLY PROCESS THE CAPTURED WIDE-AREA IMAGE DURING STEP G TO DETECT THE OCCURANCE OF HIGH SPATIAL-INTENSITY LEVELS IN THE CAPTURED WIDE-AREA IMAGE, INDICATIVE OF A SPECULAR REFLECTION CONDITION.

STEP I:

IF A SPECULAR REFLECTION CONDITION IS STILL DETECTED IN THE PROCESSED WIDE-AREA IMAGE, THEN DRIVE THE OTHER LED SUBARRAY ASSOCIATED WITH EITHER THE NEAR FIELD OR FAR FIELD WIDE AREA ILLUMINATION ARRAY.

IF A SPECULAR REFLECTION CONDITION IS NOT DETECTED IN THE PROCESSED WIDE-AREA IMAGE, THEN DRIVE USE THE DETECTED RANGE AND THE MEASURED LIGHT EXPOSURE LEVEL TO DRIVE THE SAME LED SUBARRAY (AS IN STEP C) ASSOCIATED WITH EITHER THE NEAR FIELD OR FAR FIELD WIDE AREA ILLUMINATION ARRAY.

STEP J: CAPTURE A WIDE-AREA IMAGE AT THE CMOS IMAGE SENSING ARRAY USING THE ILLUMINATION FIELD PRODUCED DURING STEP I.

STEP K: RAPIDLY PROCESS THE CAPTURED WIDE-AREA IMAGE DURING STEP J TO DETECT THE ABSENCE OF HIGH SPATIAL-INTENSITY LEVELS IN THE CAPTURED WIDE-AREA IMAGE, CONFIRMING THE ELIMINATION OF THE ONCE DETECTED SPECULAR REFLECTION CONDITION.

FIG. 13M2

STEP L:

IF NO SPECULAR REFLECTION CONDITION IS DETECTED IN THE PROCESSED WIDE-AREA IMAGE AT STEP K, THEN PROCESS THE WIDEAREA IMAGE USING MODE(S) SELECTED FOR THE MULTI-MODE IMAGEPROCESSING BAR CODE READING SUBSYSTEM.

IF A SPECULAR REFLECTION CONDITION IS STILL DETECTED IN THE PROCESSED WIDE-AREA IMAGE, THEN RETURN TO STEP A REPEAT STEPS A THROUGH K.

FIG. 13M3

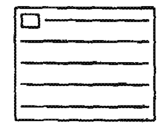
Symbologies Readable By Multi-Mode Bar Code Symbol Reading Subsystem

(1) Code 128	(2) Code 39	(3) I2of5
(4) Code93	(5) Codabar	(6) UPC/EAN
(7) Telepen	(8) UK-Plessey	(9) Trioptic
(10) Matrix 2of5	(11) Airline 2of5	(12) Straight 2of5
(13) MSI-Plessey	(14) Code11	(15) PDF417

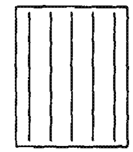
FIG. 14

Modes of Operation of Multi-mode Bar Code Reading Subsystem

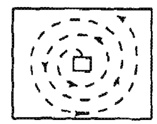
- Automatic – Look for multiple barcodes incrementally and continue looking until entire image is processed



(1) or (2)

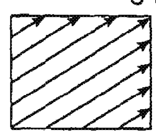


- Manual – Look for a programmable number of barcodes starting from center of image



- NoFinder – Look for one barcode in picket-fence orientation starting from center of image

- OmniScan – Look for one barcode along pre-determined orientations



- ROI-Specific Method – Look for bar code at specific region of interest (ROI) in captured image

FIG. 15

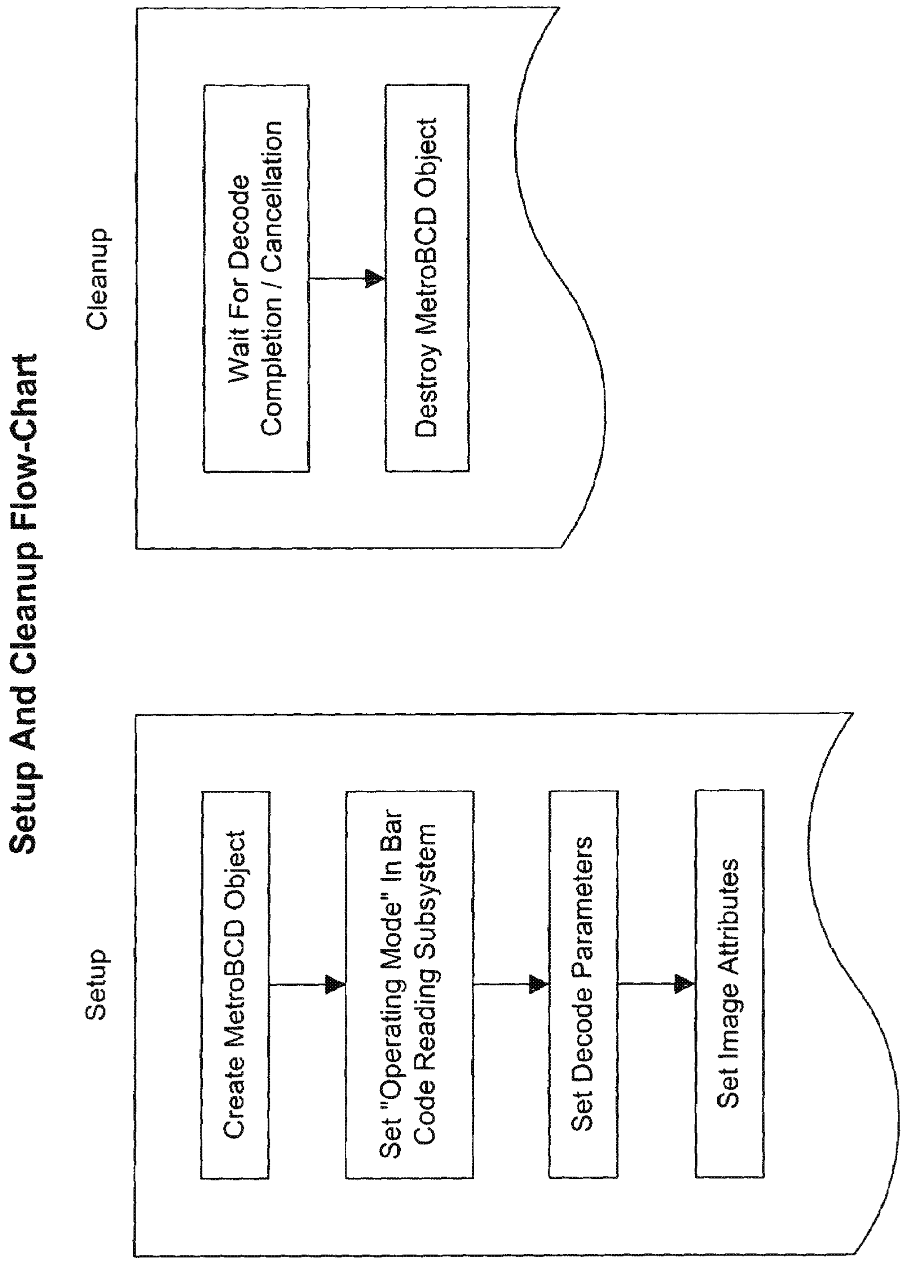

Setup And Cleanup Flow-Chart
Cleanup
Wait For Decode Completion / Cancellation
Destroy MetroBCD Object
Setup
Create MetroBCD Object
Set "Operating Mode" In Bar Code Reading Subsystem
Set Decode Parameters
Set Image Attributes

FIG. 16

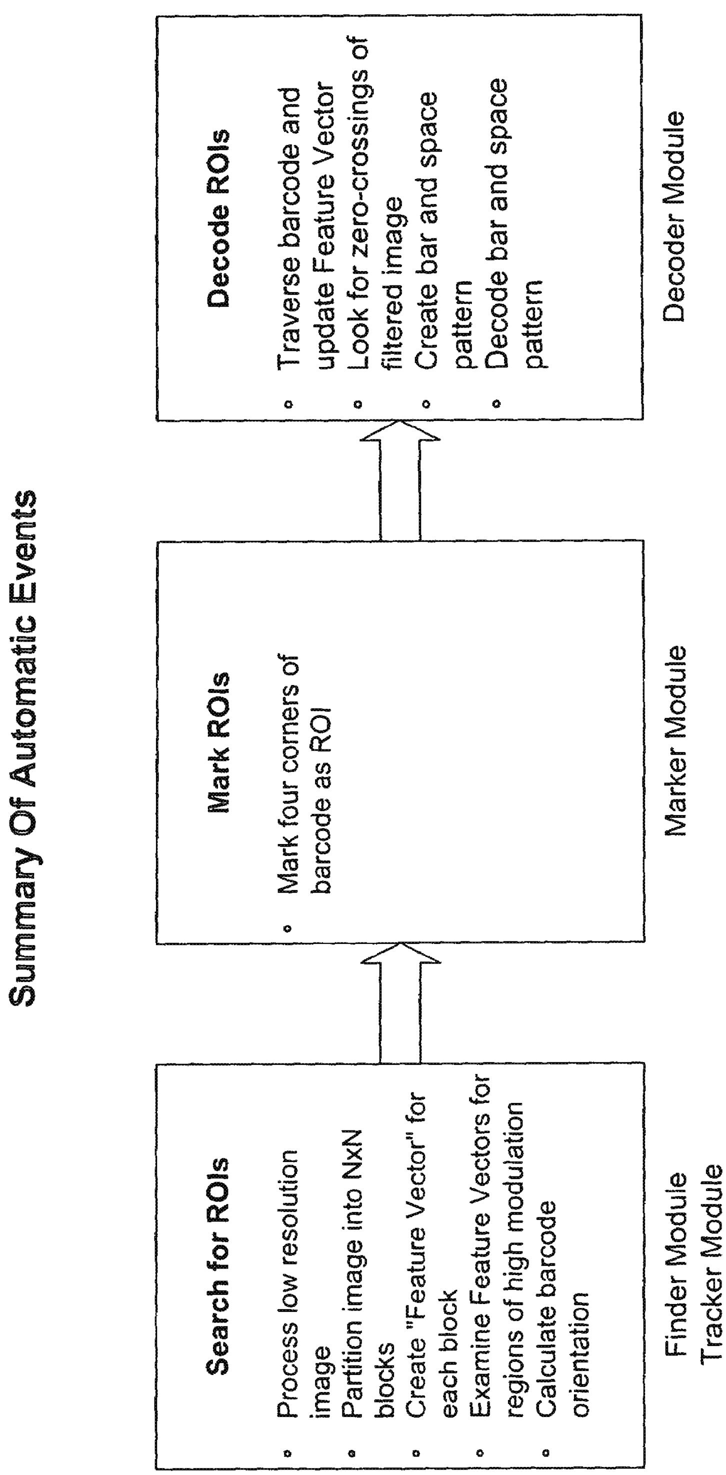

Summary Of Automatic Events
Search for ROIs
Process low resolution image
Partition image into NxN blocks
Create "Feature Vector" for each block
Examine Feature Vectors for regions of high modulation
Calculate barcode orientation
Finder Module
Tracker Module
Mark ROIs
Mark four corners of barcode as ROI
Marker Module
Decode ROIs
Traverse barcode and update Feature Vector
Look for zero-crossings of filtered image
Create bar and space pattern
Decode bar and space pattern
Decoder Module

FIG. 17A

Automatic Mode Flow-Chart

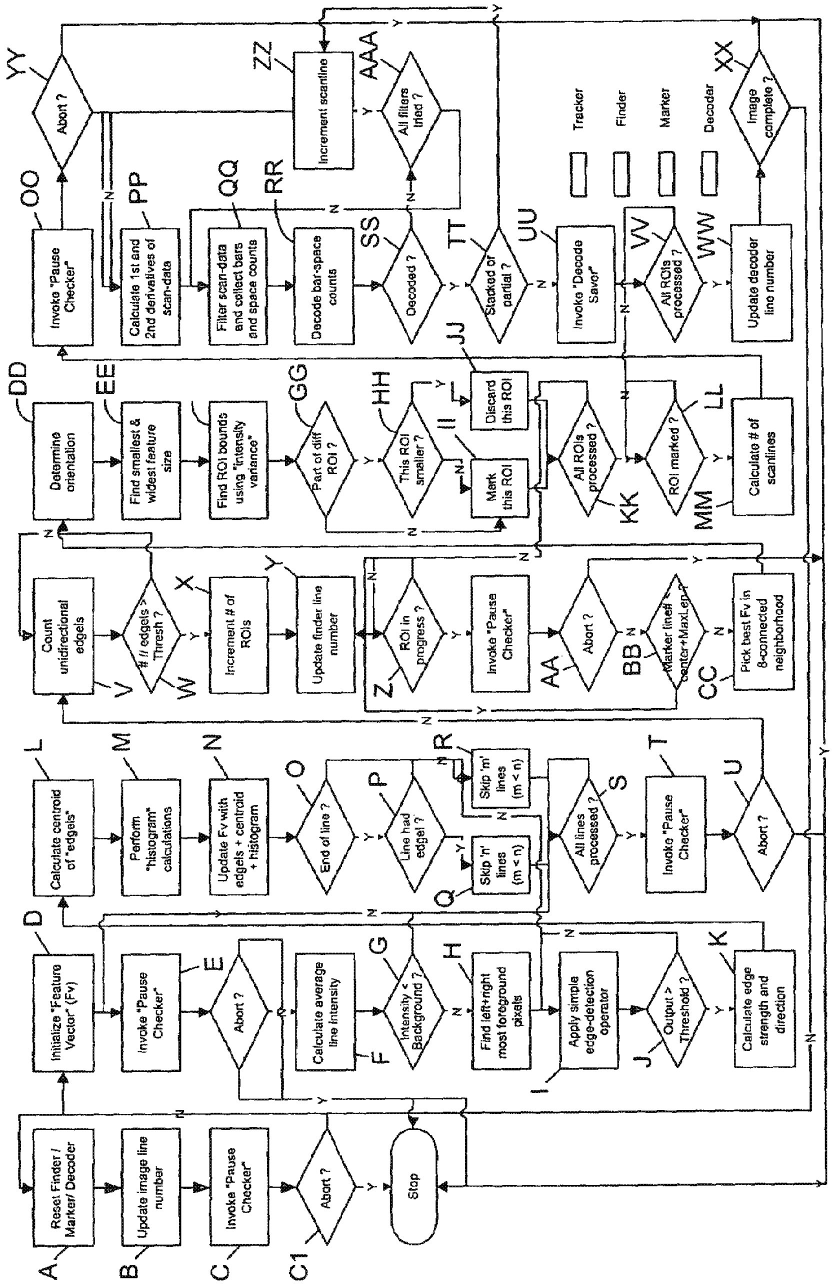


FIG. 17B

HONEYWELL-00192933

JA2634

Step 1: Search for ROIs: Low resolution processing

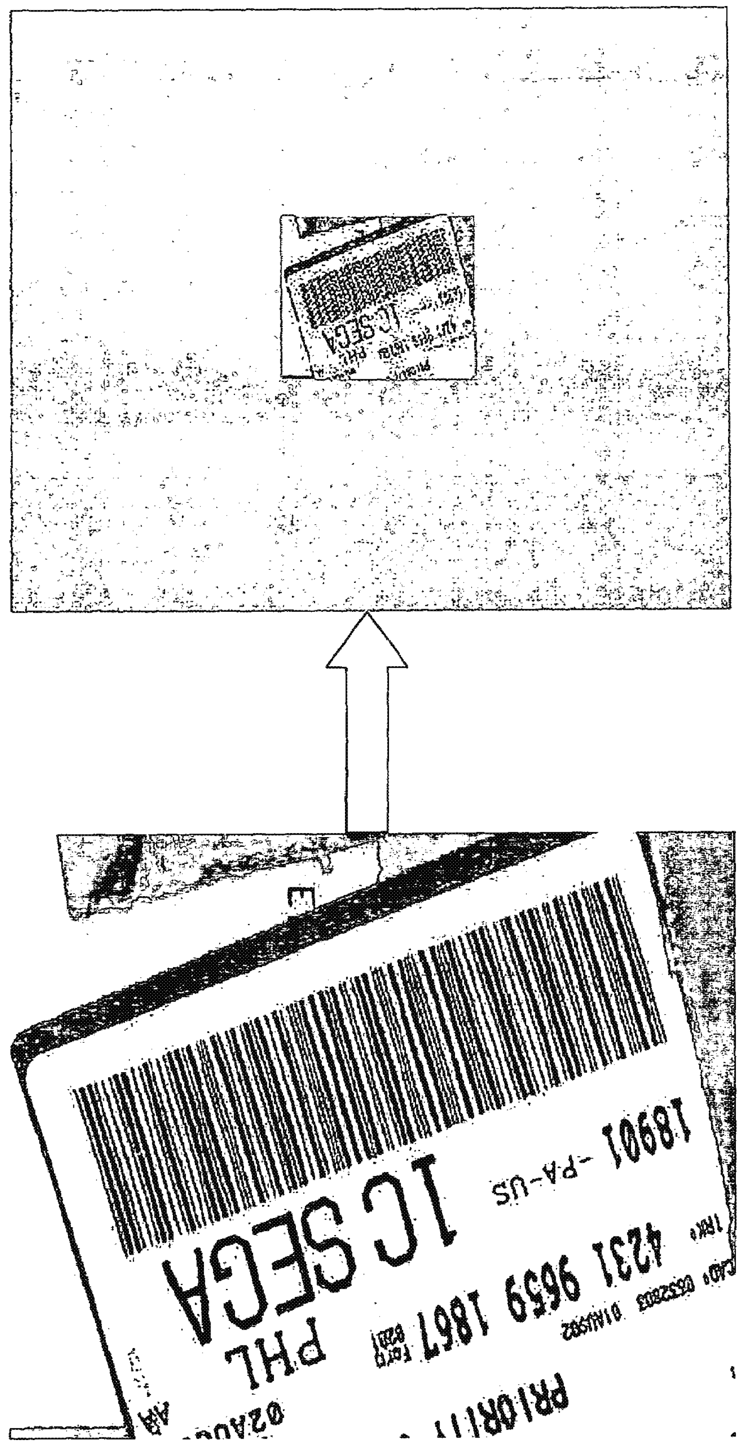
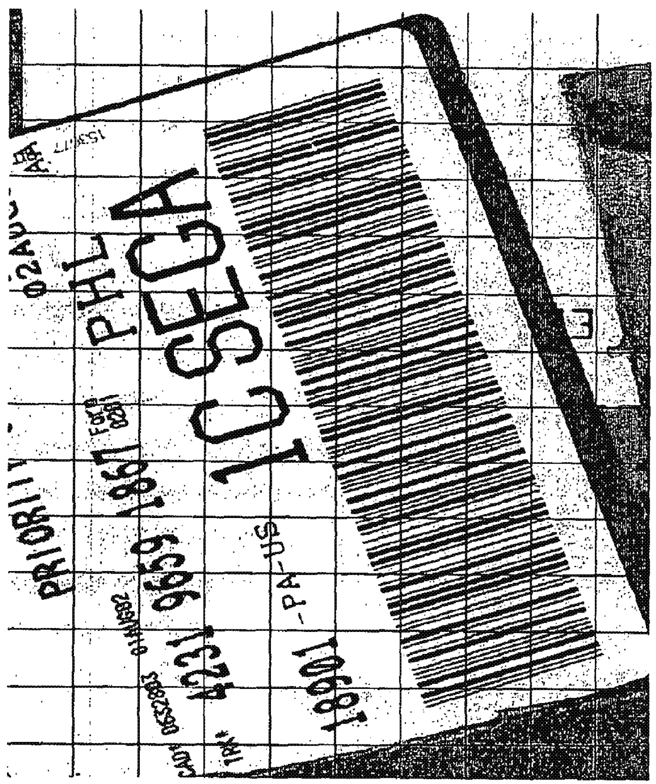


FIG. 18A

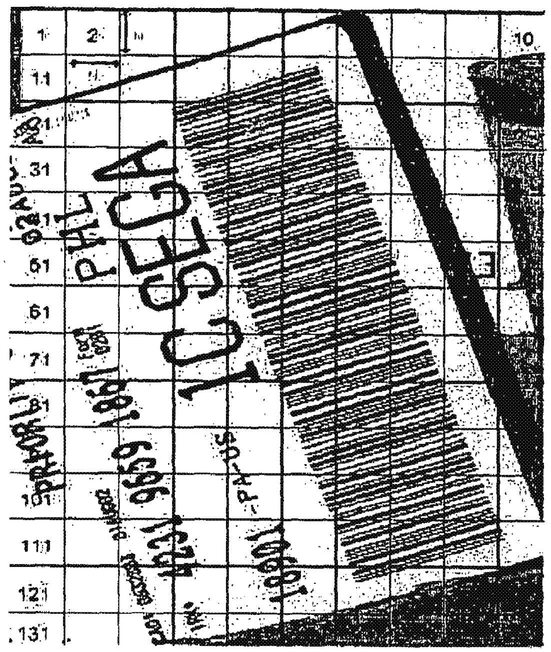
Step 2: Search for ROIs: Partition image



- Image overlaid with XY grid
- Each block formed by grids has an associated "feature vector" (Fv)
- Feature vectors are analyzed for the presence of parallel lines
- All feature vector calculations are performed on the low-resolution image

FIG. 18B

Step 3: Search for ROIs: Create feature vectors

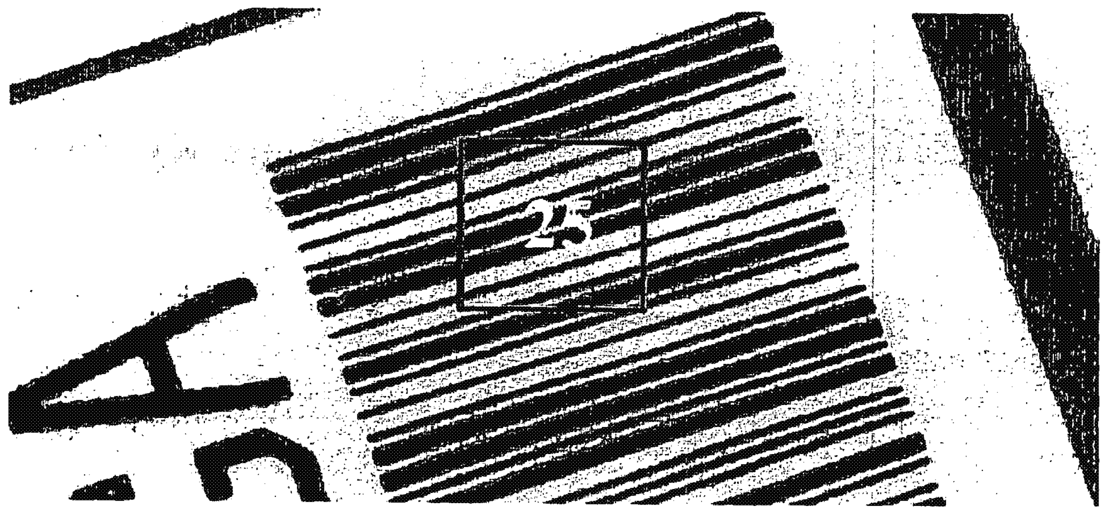


Fv =

- Gradient vectors
- Edge density
- Number of parallel edge vectors
- Centroid of edgels
- Intensity variance
- Histogram of intensities

FIG. 18C

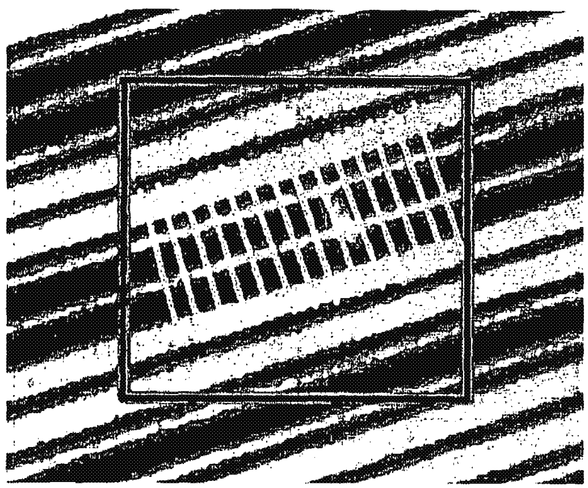
Step 4: Mark ROIs: Examine feature vectors



- High edge density
- Large number of parallel edge vectors
- Large intensity variance

FIG. 18D

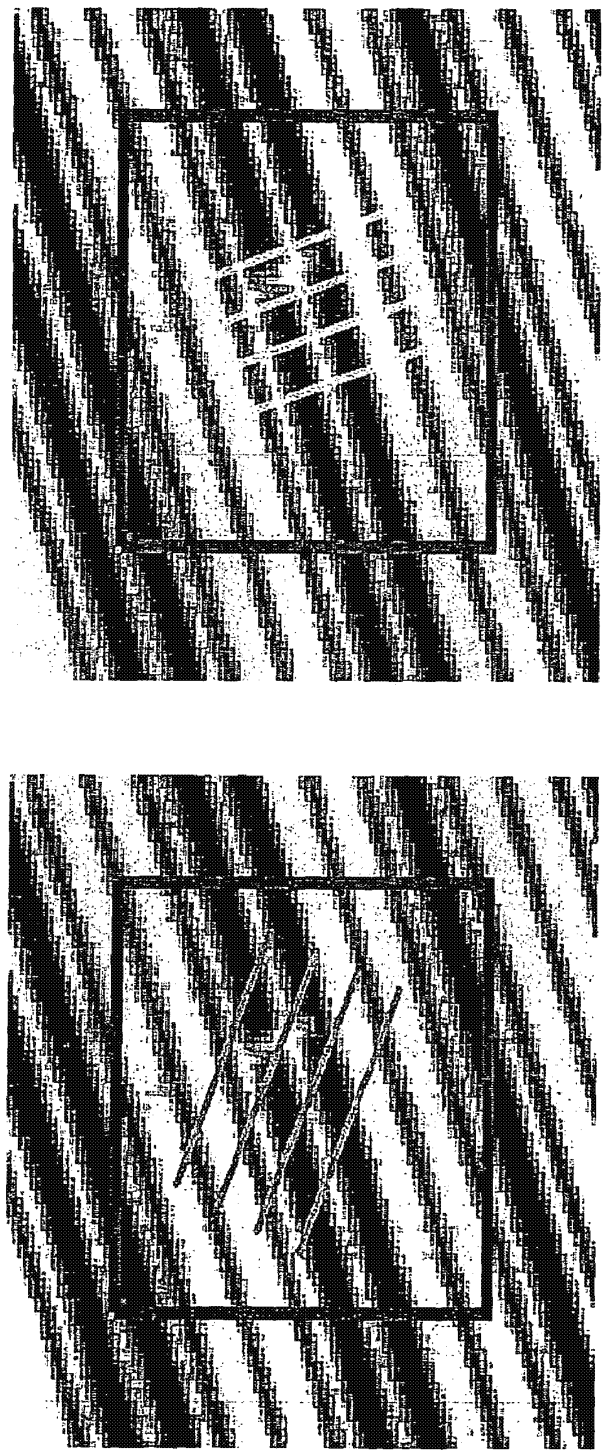
Step 5: Mark ROIs: Calculate barcode orientation



- Within each "feature vector" block the barcode is traversed ("sliced") at different angles
- The slices are matched with each other based on "least mean square error"
- The correct orientation is that angle that matches in a "mean square error" sense every slice of the barcode

FIG. 18E

Step 5: Mark ROIs: Calculate barcode orientation



High mean square error between slices

Lowest mean square error between slices - Correct orientation

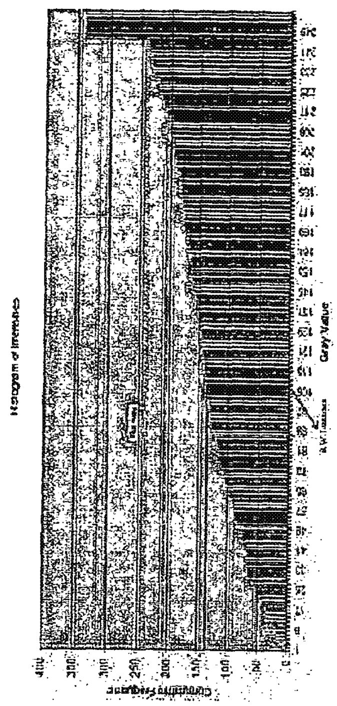
FIG. 18F

Step 6: Mark ROIs: Mark four corners of barcode

- From here on all operations are performed on the full-resolution image
- Barcode is traversed in either direction starting from center of block
- Using intensity variance the extent of modulation is detected (1 & 2)
- Starting from 1 & 2 and moving perpendicular to barcode orientation the four corners are determined (3, 4, 5, 6)
- 3, 4, 5, 6 define the ROI

FIG. 18G

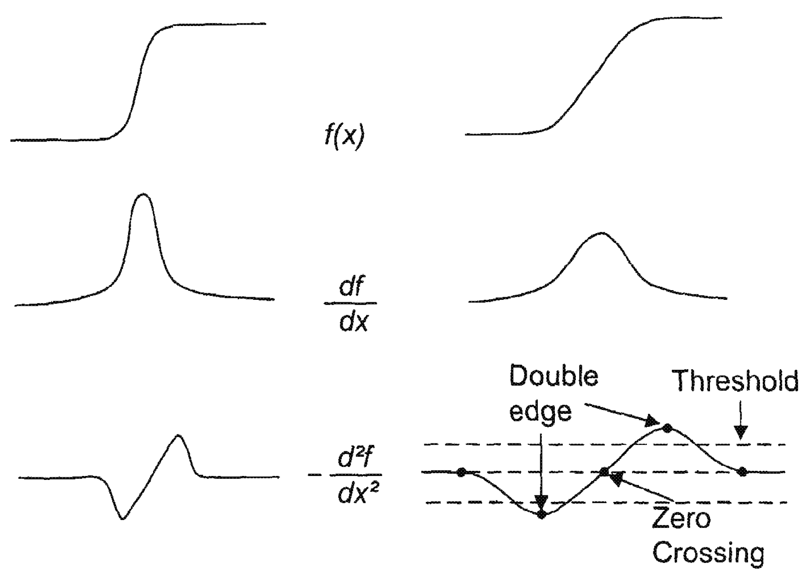
Step 7: Decode ROIs: Update feature vectors



- Histogram component of Fv is updated while traversing barcode
- Estimate of Black-to-White transition is calculated
- Estimate of narrow & wide elements are calculated

FIG. 18H

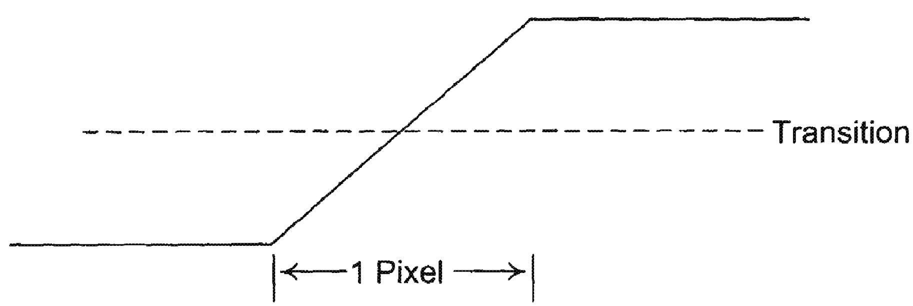
Step 8: Decode ROIs: Look for zero-crossings



- Barcode image is median filtered in a direction perpendicular to barcode orientation
- Second derivative zero-crossings define edge transitions
- Zero-crossing data used only for detecting the edge transitions
- B/W transition estimates put upper and lower bounds to bar and space gray levels

FIG. 18I

Step 9: Decode ROIs: Create bar and space pattern



- Edge transition is modeled as a ramp
- Edge transition is assumed to be 1-pixel wide
- Edge transition location is determined at the sub-pixel level
- Bar and space counts are gathered using edge transition data

FIG. 18J

Step 10: Decode ROIs: Decode bar and space pattern

- Bar and space data framed with “borders”
- Bar and space data decoded using existing Metrologic laser-scanner algorithms

FIG. 18K

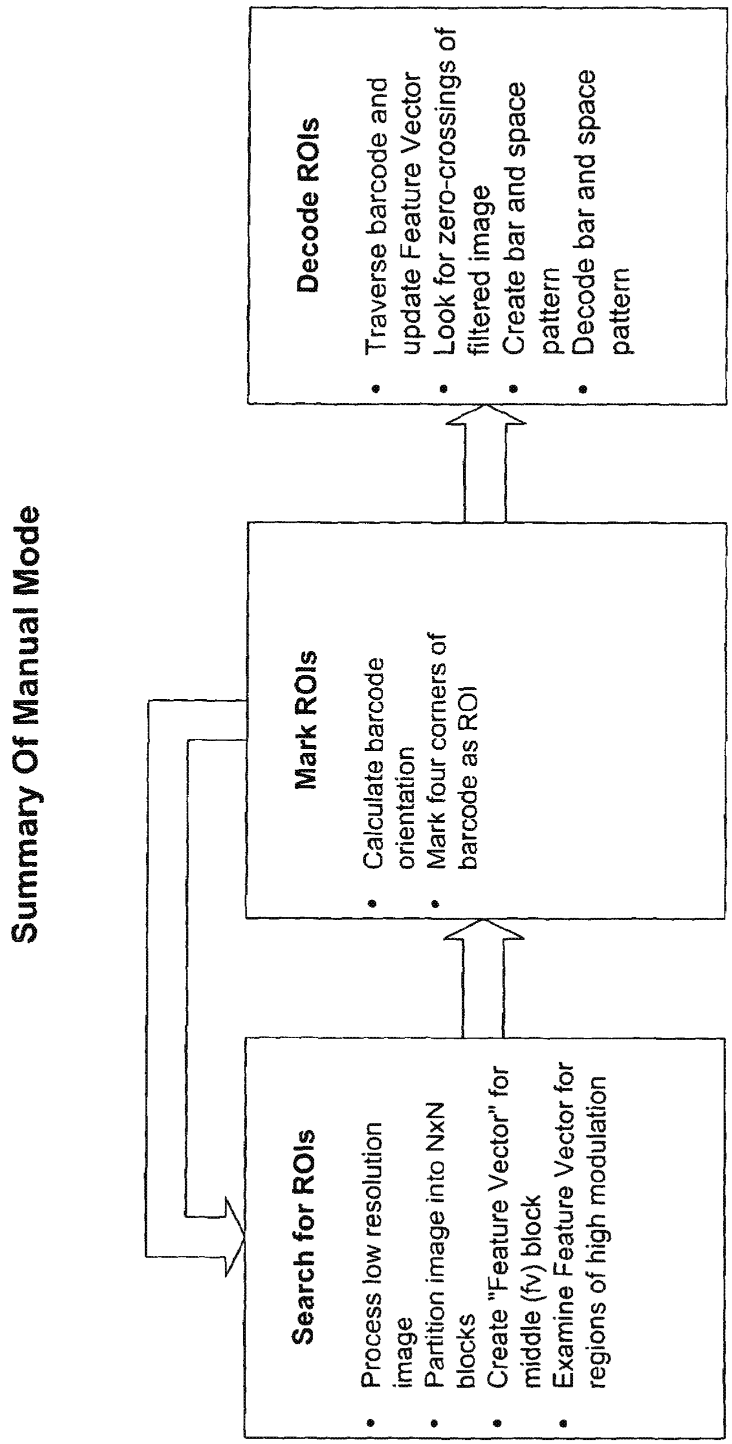

Summary Of Manual Mode
Search for ROIs
• Process low resolution image
• Partition image into NxN blocks
• Create "Feature Vector" for middle (fv) block
• Examine Feature Vector for regions of high modulation
Mark ROIs
• Calculate barcode orientation
• Mark four corners of barcode as ROI
Decode ROIs
• Traverse barcode and update Feature Vector
• Look for zero-crossings of filtered image
• Create bar and space pattern
• Decode bar and space pattern

FIG. 19A

Manual Mode Flow-Chart

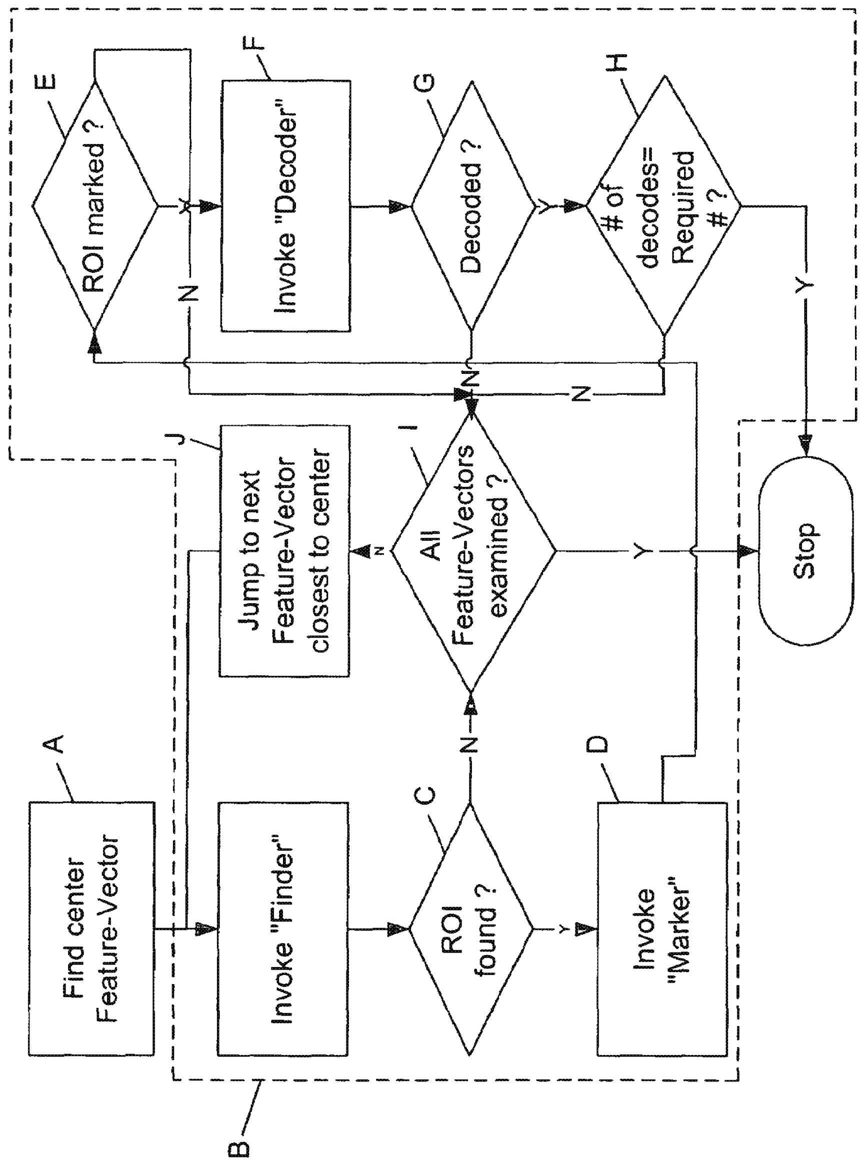


FIG. 19B

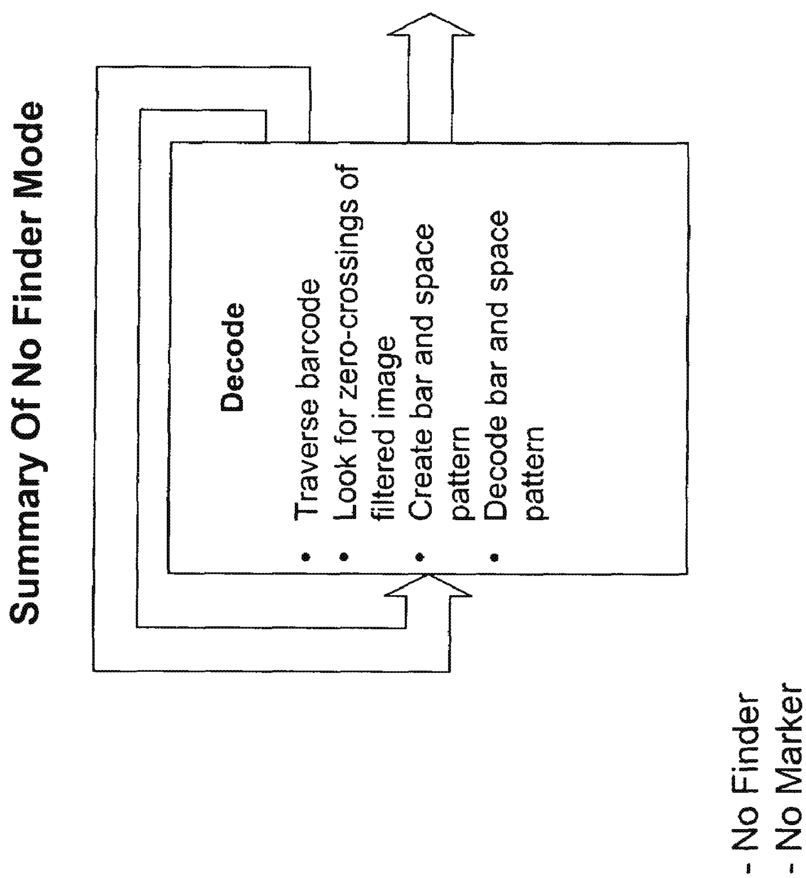

Summary Of No Finder Mode
Decode
• Traverse barcode
• Look for zero-crossings of filtered image
• Create bar and space pattern
• Decode bar and space pattern
- No Finder
- No Marker

FIG. 20A

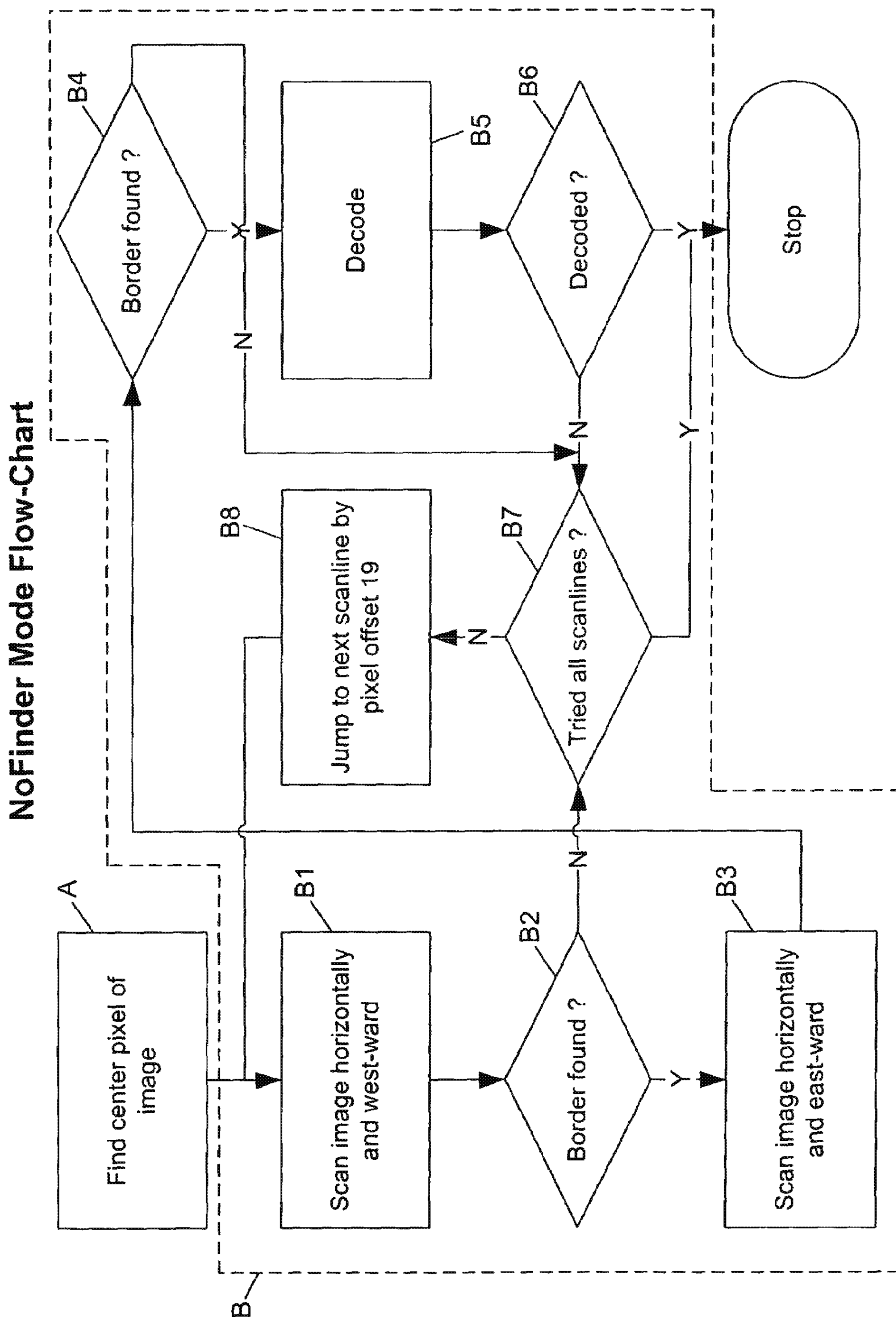

NoFinder Mode Flow-Chart
A
Find center pixel of image
B
B1
Scan image horizontally and west-ward
B2
Border found ?
Y
N
B3
Scan image horizontally and east-ward
B4
Border found ?
Y
N
B5
Decode
B6
Decoded ?
Y
N
Stop
B7
Tried all scanlines ?
Y
N
B8
Jump to next scanline by pixel offset 19

FIG. 20B

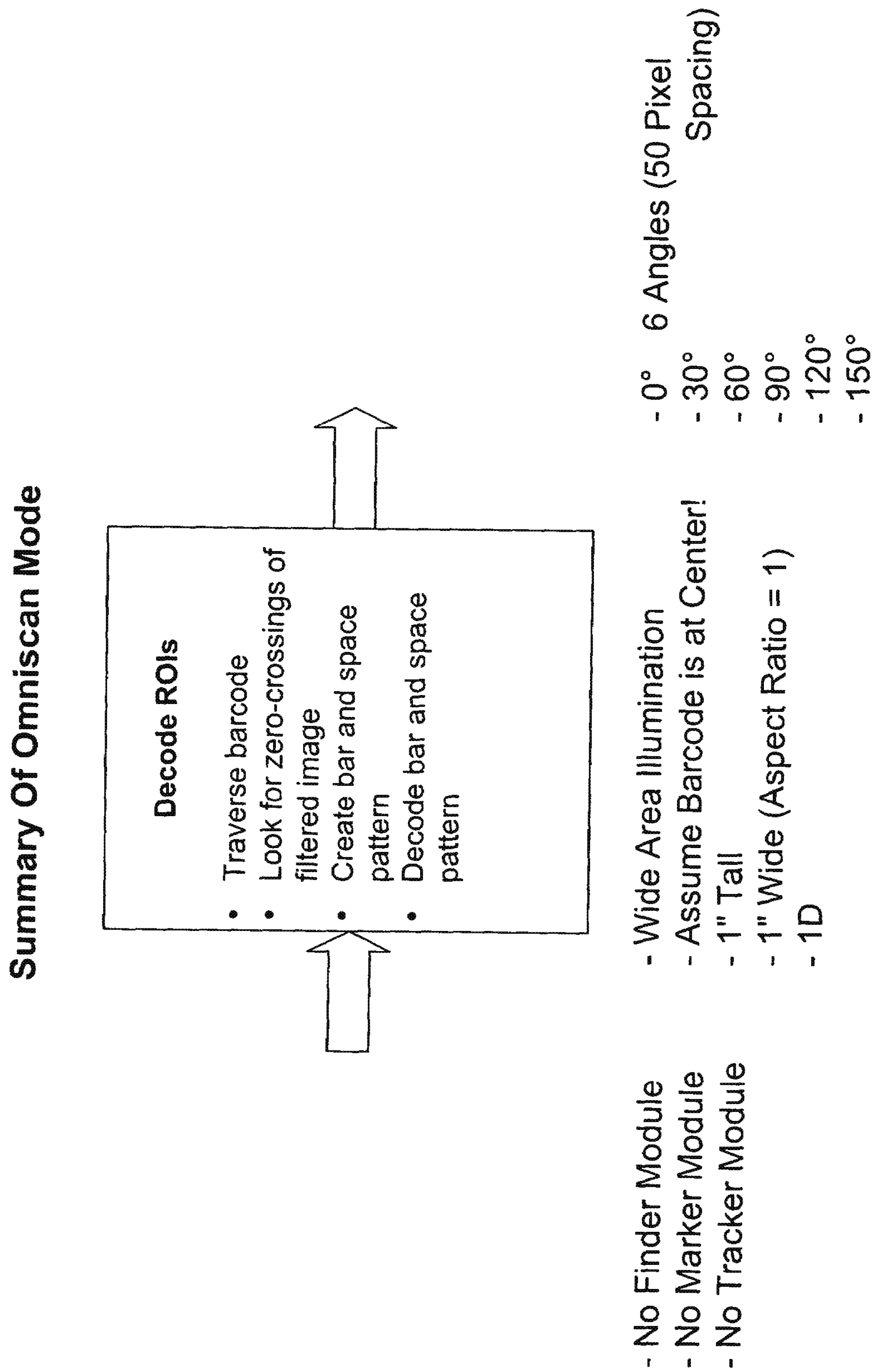

Summary Of Omniscan Mode
Decode ROIs
• Traverse barcode
• Look for zero-crossings of filtered image
• Create bar and space pattern
• Decode bar and space pattern
- No Finder Module
- No Marker Module
- No Tracker Module
- Wide Area Illumination
- Assume Barcode is at Center!
- 1" Tall
- 1" Wide (Aspect Ratio = 1)
- 1D
- 0° 6 Angles (50 Pixel Spacing)
- 30°
- 60°
- 90°
- 120°
- 150°

FIG. 21A

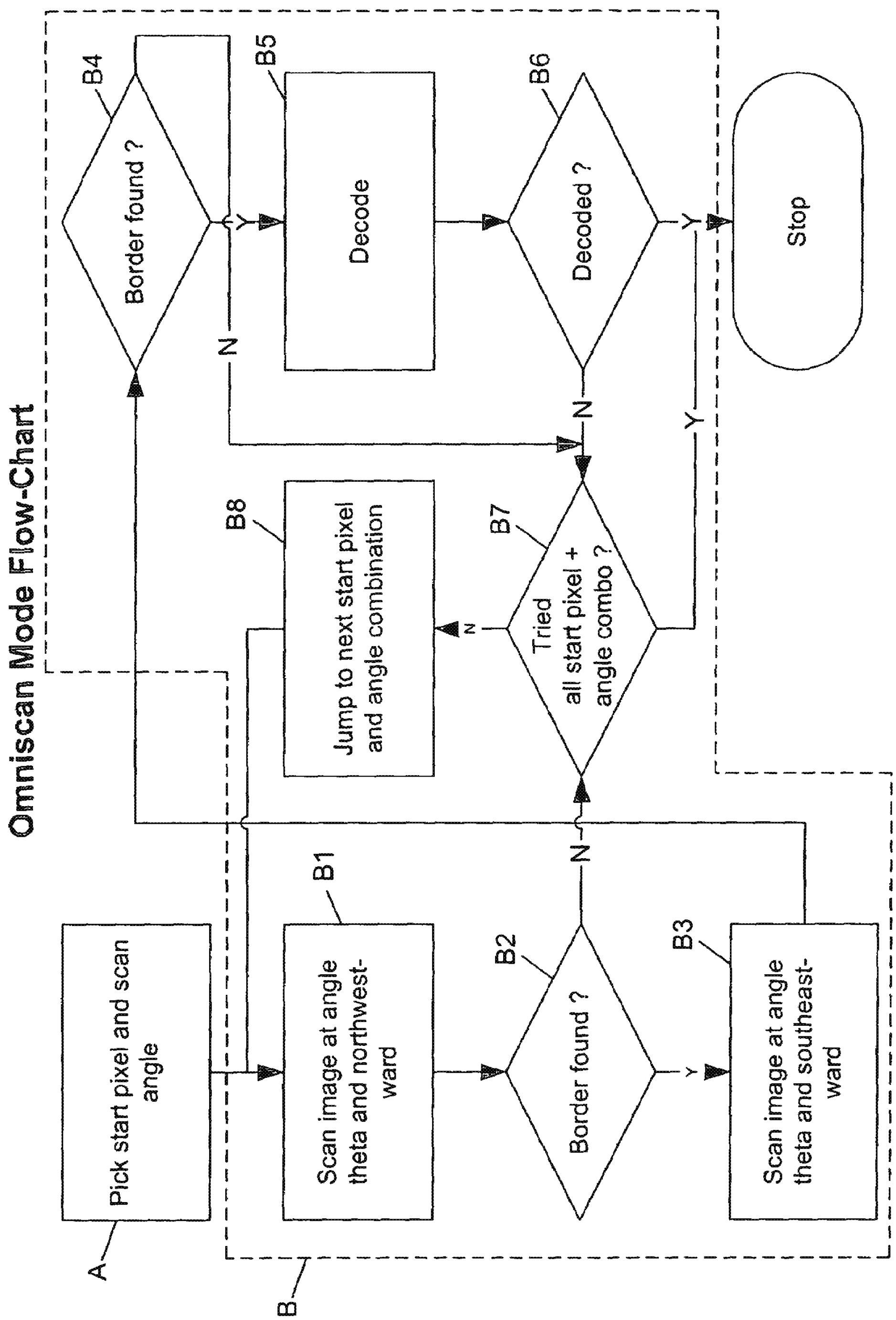

Omniscan Mode Flow-Chart
Pick start pixel and scan angle
A
Scan image at angle theta and northwest-ward
B1
Border found ?
B2
Y
N
Scan image at angle theta and southeast-ward
B3
B
Border found ?
B4
Y
N
Decode
B5
Decoded ?
B6
Y
N
Stop
Tried all start pixel + angle combo ?
B7
Y
N
Jump to next start pixel and angle combination
B8

FIG. 21B

Summary Of ROI-Specific Mode

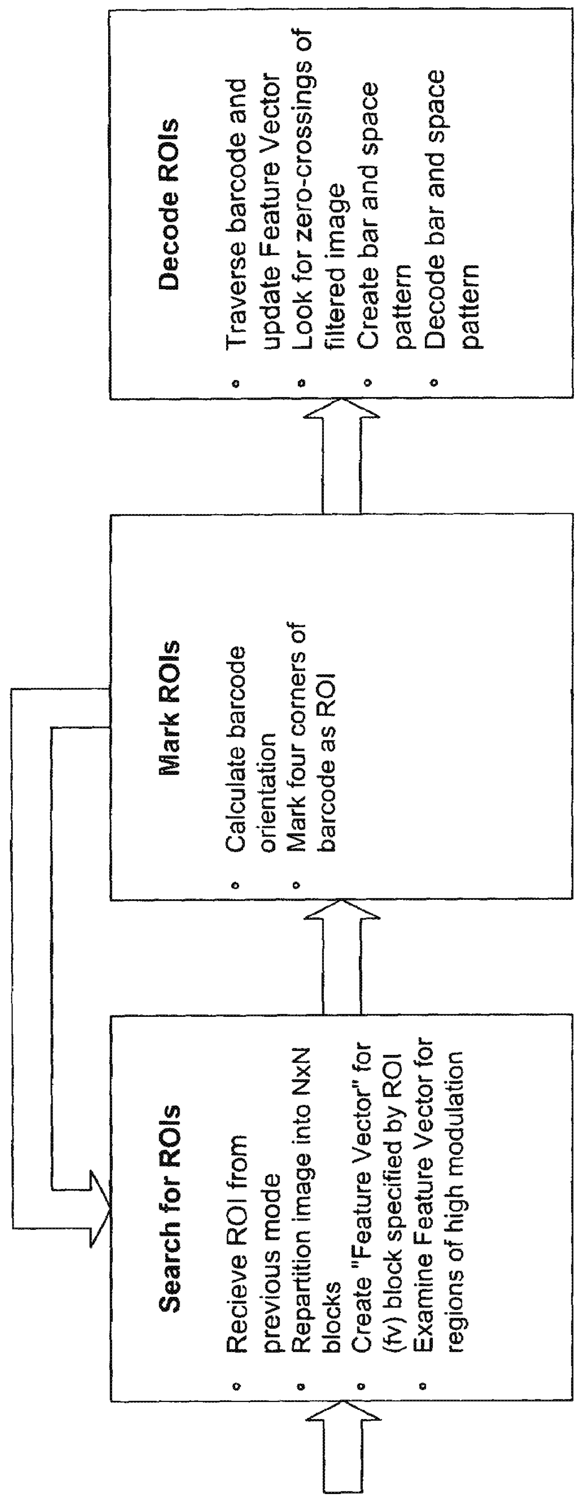


FIG. 22A

ROI-Specific Mode Flow-Chart

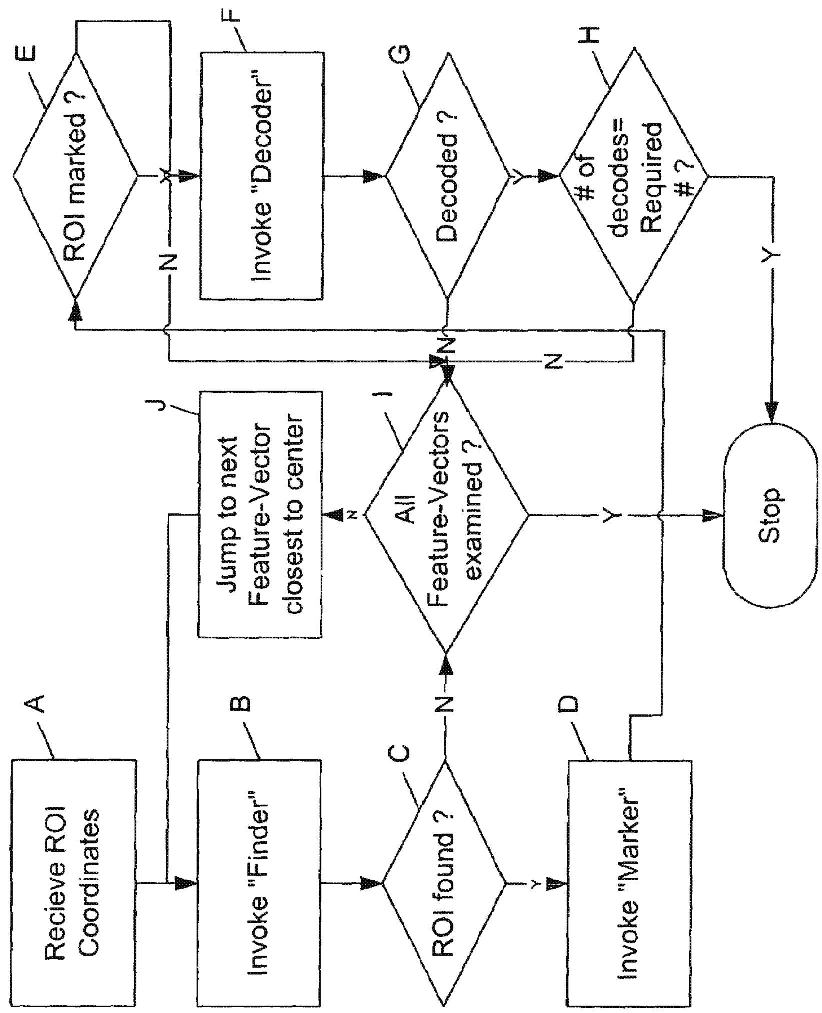


FIG. 22B

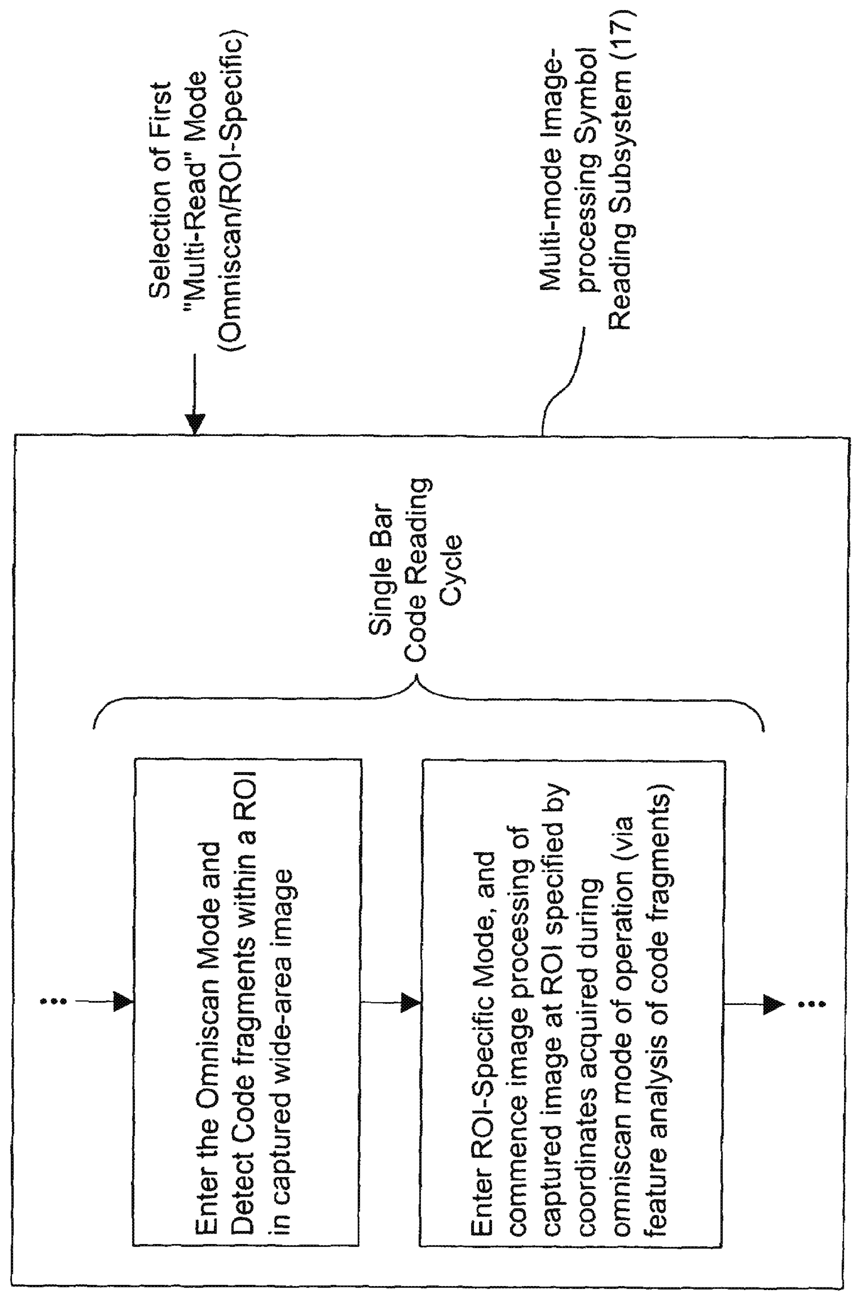

Selection of First "Multi-Read" Mode (Omniscan/ROI-Specific)
Multi-mode Image-processing Symbol Reading Subsystem (17)
Single Bar Code Reading Cycle
Enter the Omniscan Mode and Detect Code fragments within a ROI in captured wide-area image
Enter ROI-Specific Mode, and commence image processing of captured image at ROI specified by coordinates acquired during omniscan mode of operation (via feature analysis of code fragments)

FIG. 23

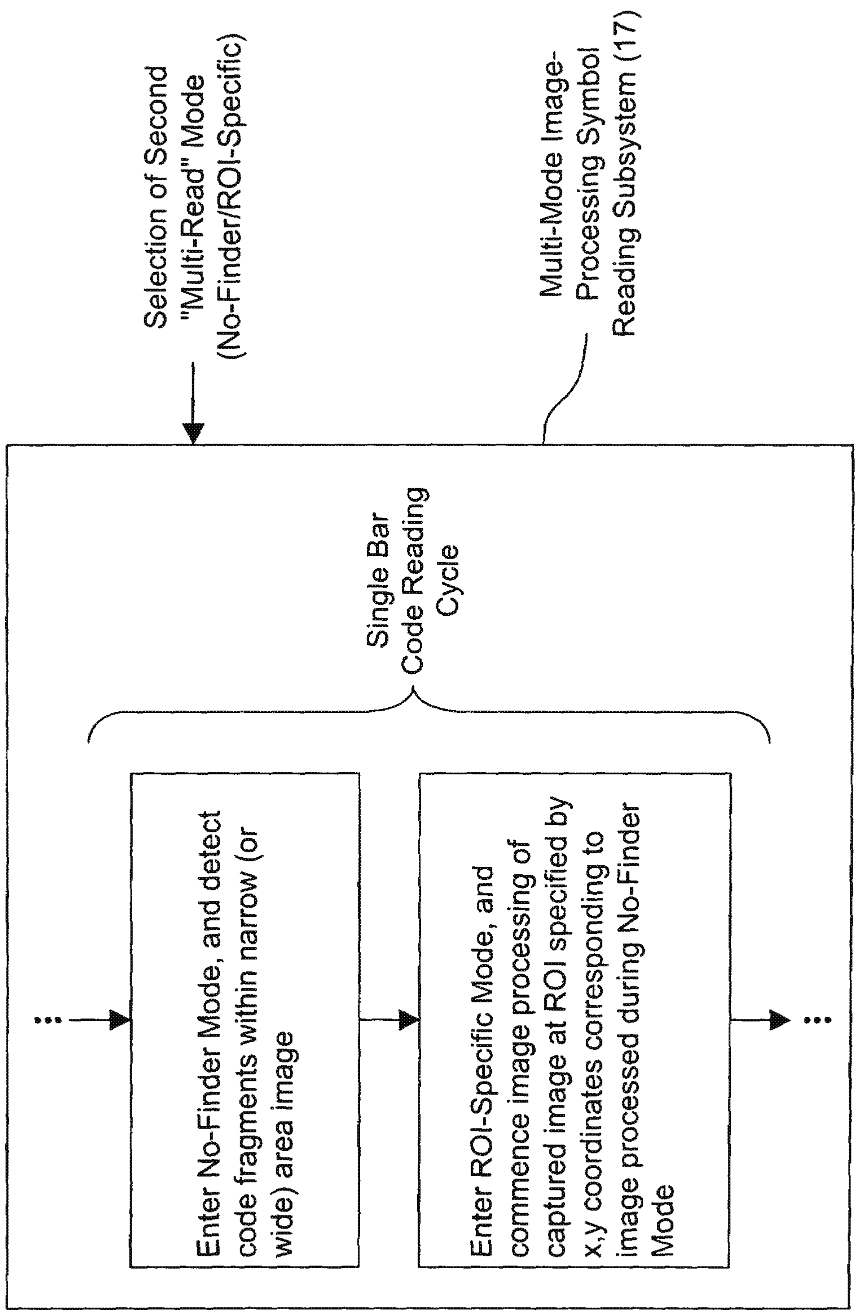

Selection of Second "Multi-Read" Mode (No-Finder/ROI-Specific)
Multi-Mode Image-Processing Symbol Reading Subsystem (17)
Single Bar Code Reading Cycle
Enter No-Finder Mode, and detect code fragments within narrow (or wide) area image
Enter ROI-Specific Mode, and commence image processing of captured image at ROI specified by x,y coordinates corresponding to image processed during No-Finder Mode

FIG. 24

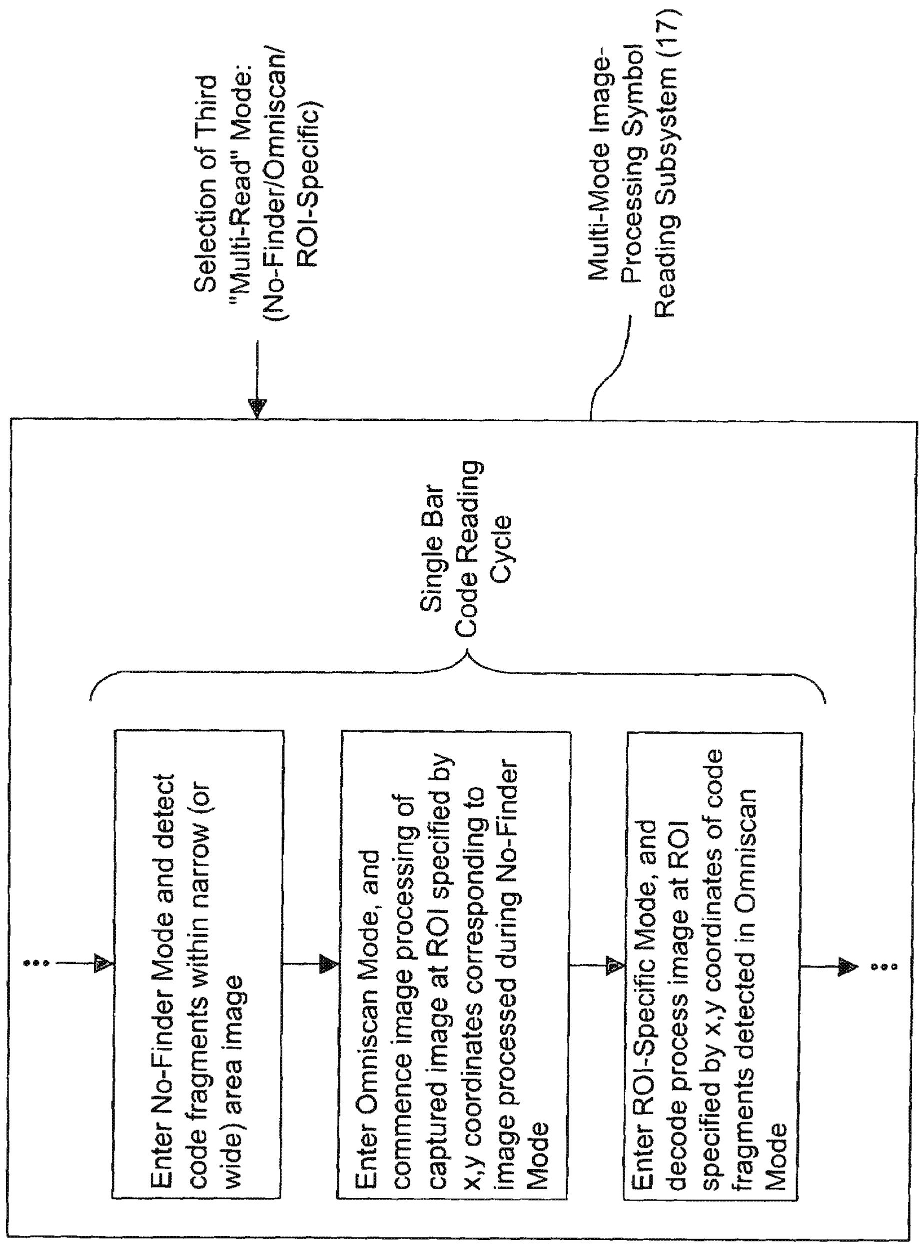

Selection of Third "Multi-Read" Mode: (No-Finder/Omniscan/ ROI-Specific)
Multi-Mode Image-Processing Symbol Reading Subsystem (17)
Single Bar Code Reading Cycle
Enter No-Finder Mode and detect code fragments within narrow (or wide) area image
Enter Omniscan Mode, and commence image processing of captured image at ROI specified by x,y coordinates corresponding to image processed during No-Finder Mode
Enter ROI-Specific Mode, and decode process image at ROI specified by x,y coordinates of code fragments detected in Omniscan Mode

FIG. 25

PROGRAMMABLE MODES OF BAR CODE SYMBOL READING OPERATION WITHIN THE HAND-SUPPORTABLE DIGITAL IMAGING-BASED BAR CODE SYMBOL READER OF THE PRESENT INVENTION

Programmed Mode of System Operation No.1: Manually-Triggered Single-Attempt 1D Single-Read Mode Employing the No-Finder Mode of Operation

Programmed Mode of System Operation No.2: Manually-Triggered Multiple-Attempt 1D Single-Read Mode Employing the No-Finder Mode of Operation

Programmed Mode of System Operation No.3: Manually-Triggered Single-Attempt 1D/2D Single-Read Mode Employing the No-Finder And The Automatic Or Manual Modes of Operation

Programmed Mode of System Operation No.4: Manually-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing the No-Finder And The Automatic Or Manual Modes of Operation

Programmed Mode of System Operation No.5: Manually-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing the No-Finder And The Automatic Or Manual Modes of Operation

Programmed Mode of System Operation No.6: Automatically- Triggered Single-Attempt 1D Single-Read Mode Employing The No-Finder Mode Of Operation

Programmed Mode of System Operation No.7: Automatically-Triggered Multi-Attempt 1D Single-Read Mode Employing The No-Finder Mode Of Operation

Programmed Mode of System Operation No.8: Automatically-Triggered Multi-Attempt 1D/2D Single-Read Mode Employing The No-Finder and Manual and/or Automatic Modes Of Operation

Programmed Mode of System Operation No.9: Automatically-Triggered Multi-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder and Manual and/or Automatic Modes Of Operation

Programmable Mode of System Operation No. 10: Automatically-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing The Manual, Automatic or Omniscan Modes Of Operation

Programmed Mode of System Operation No. 11: Semi-Automatic-Triggered Single-Attempt 1D/2D Single-Read Mode Employing The No-Finder And The Automatic Or Manual Modes Of Operation

FIG. 26A

Programmable Mode of System Operation No. 12: Semi-Automatic-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing The No-Finder And The Automatic Or Manual Modes Of Operation

Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder And The Automatic Or Manual Modes Of Decoder Operation; Programmable Mode of Operation No. 13

Programmable Mode of Operation No. 14: Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder And The Omniscan Modes Of Operation

Programmable Mode of Operation No. 15: Continuously-Automatically-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The Automatic, Manual Or Omniscan Modes Of Operation

Programmable Mode of System Operation No. 16: Diagnostic Mode Of Imaging-Based Bar Code Reader Operation

Programmable Mode of System Operation No. 17: Live Video Mode Of Imaging-Based Bar Code Reader Operation

FIG. 26B

Imaging-Based Bar Code Symbol Reading System
With Extended Multi-Mode Illumination Subsystem

- Four Modes Of Illumination
  - (1) Wide-Area For "Near" Object (0 mm-100 mm)
  - (2) Wide-Area For "Far" Object (100 mm-200 mm)
  - (3) Narrow-Area For "Near" Object (0 mm-100 mm)
  - (4) Narrow-Area For "Far" Object (100 mm-200 mm)

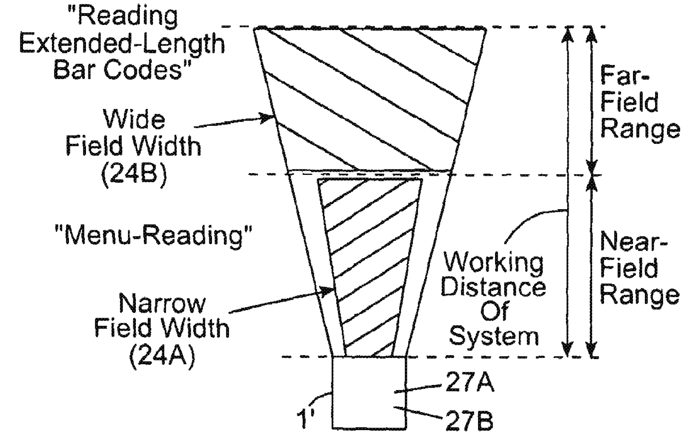


FIG. 27B

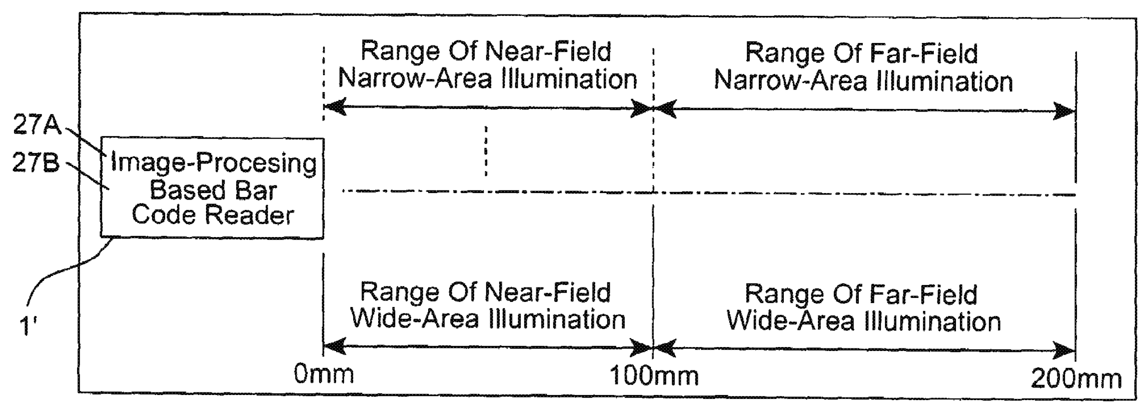
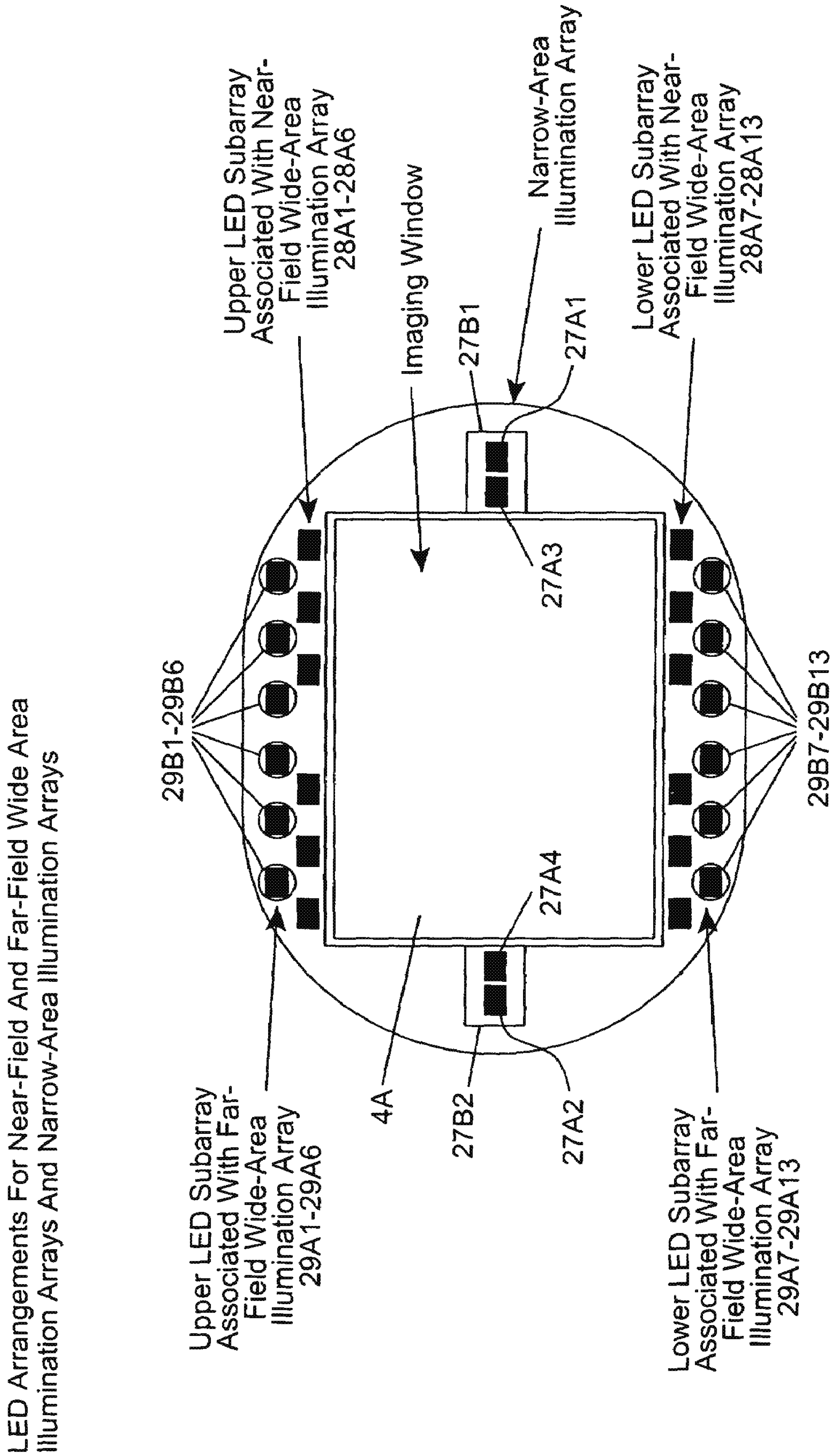


FIG. 27A



LED Arrangements For Near-Field And Far-Field Wide Area Illumination Arrays And Narrow-Area Illumination Arrays
Upper LED Subarray Associated With Near-Field Wide-Area Illumination Array 28A1-28A6
Imaging Window
27B1
Narrow-Area Illumination Array
27A1
Lower LED Subarray Associated With Near-Field Wide-Area Illumination Array 28A7-28A13
29B1-29B6
27A3
29B7-29B13
27A4
Upper LED Subarray Associated With Far-Field Wide-Area Illumination Array 29A1-29A6
4A
27B2
27A2
Lower LED Subarray Associated With Far-Field Wide-Area Illumination Array 29A7-29A13

FIG. 28

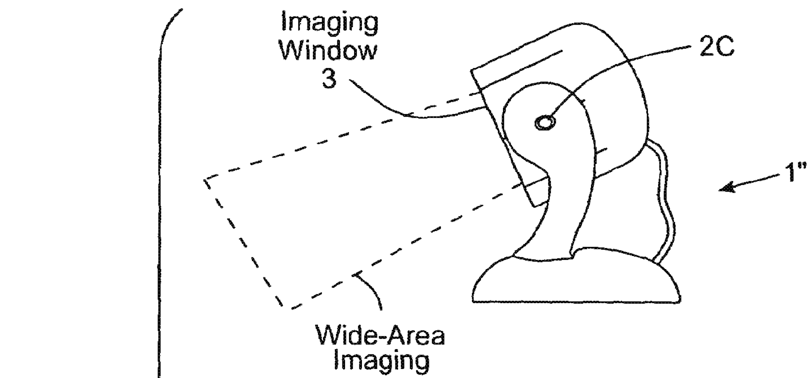


FIG. 29A

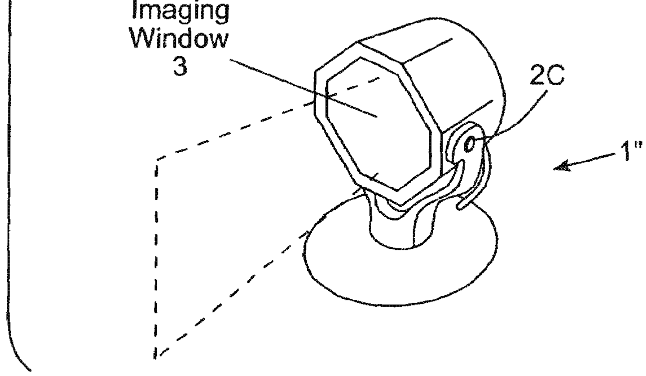


FIG. 29B

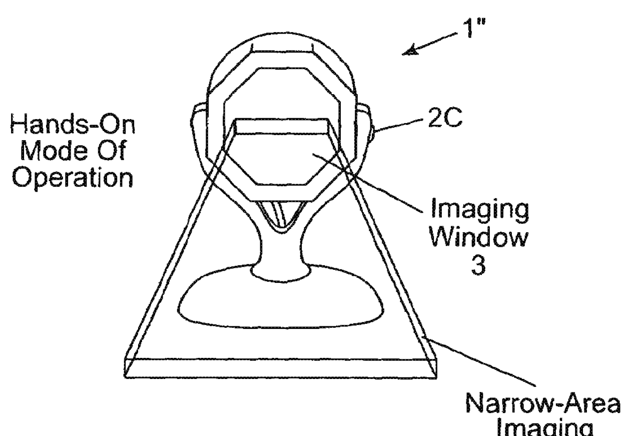


FIG. 29C

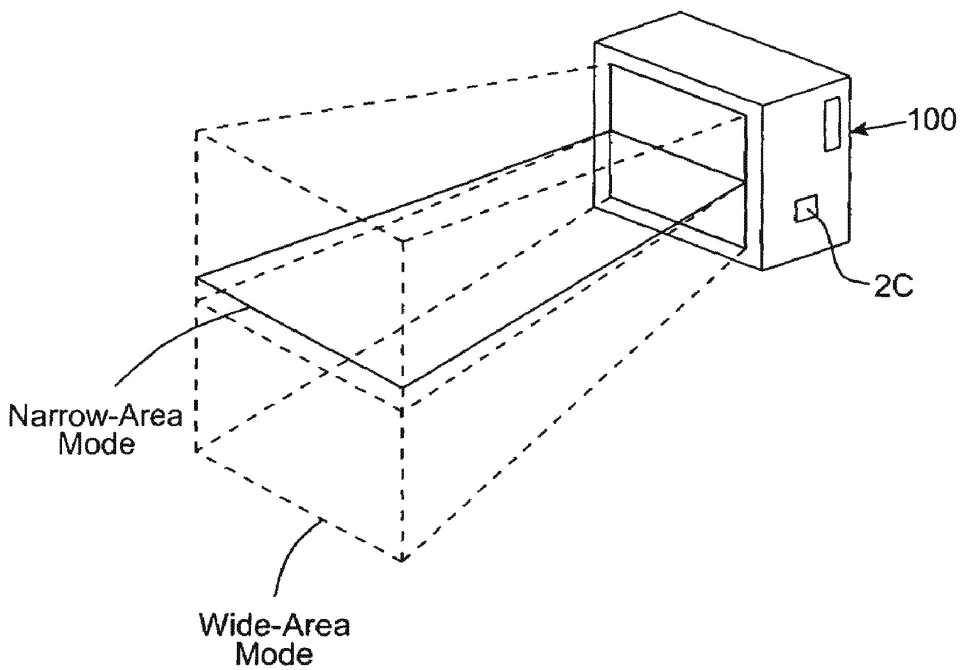


FIG. 30

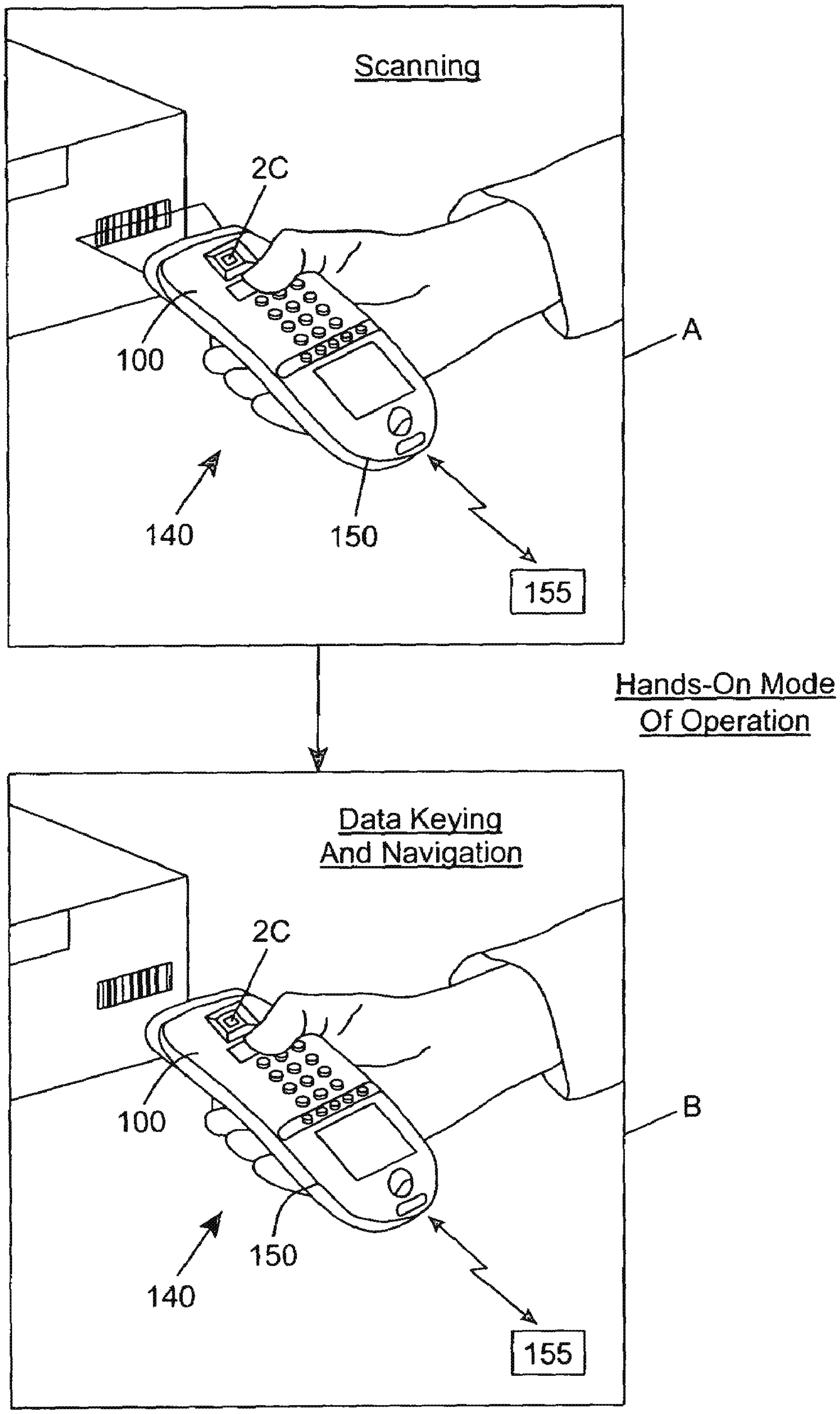

Scanning
2C
100
140
150
155
A
Hands-On Mode
Of Operation
Data Keying
And Navigation
2C
100
150
140
155
B

FIG. 31

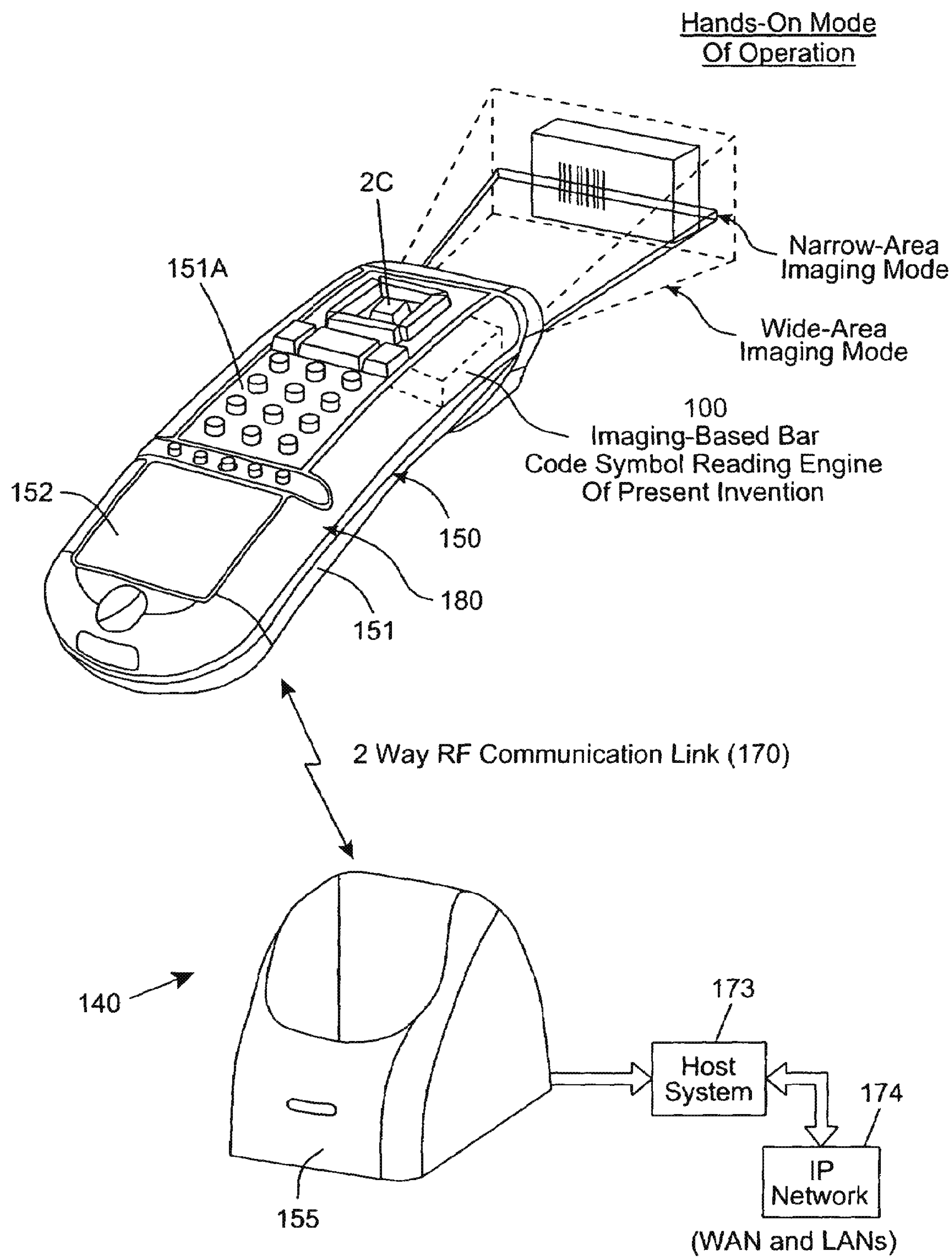

Hands-On Mode
Of Operation
2C
151A
Narrow-Area
Imaging Mode
Wide-Area
Imaging Mode
100
Imaging-Based Bar
Code Symbol Reading Engine
Of Present Invention
152
150
180
151
2 Way RF Communication Link (170)
140
173
Host
System
174
IP
Network
(WAN and LANs)
155

FIG. 32

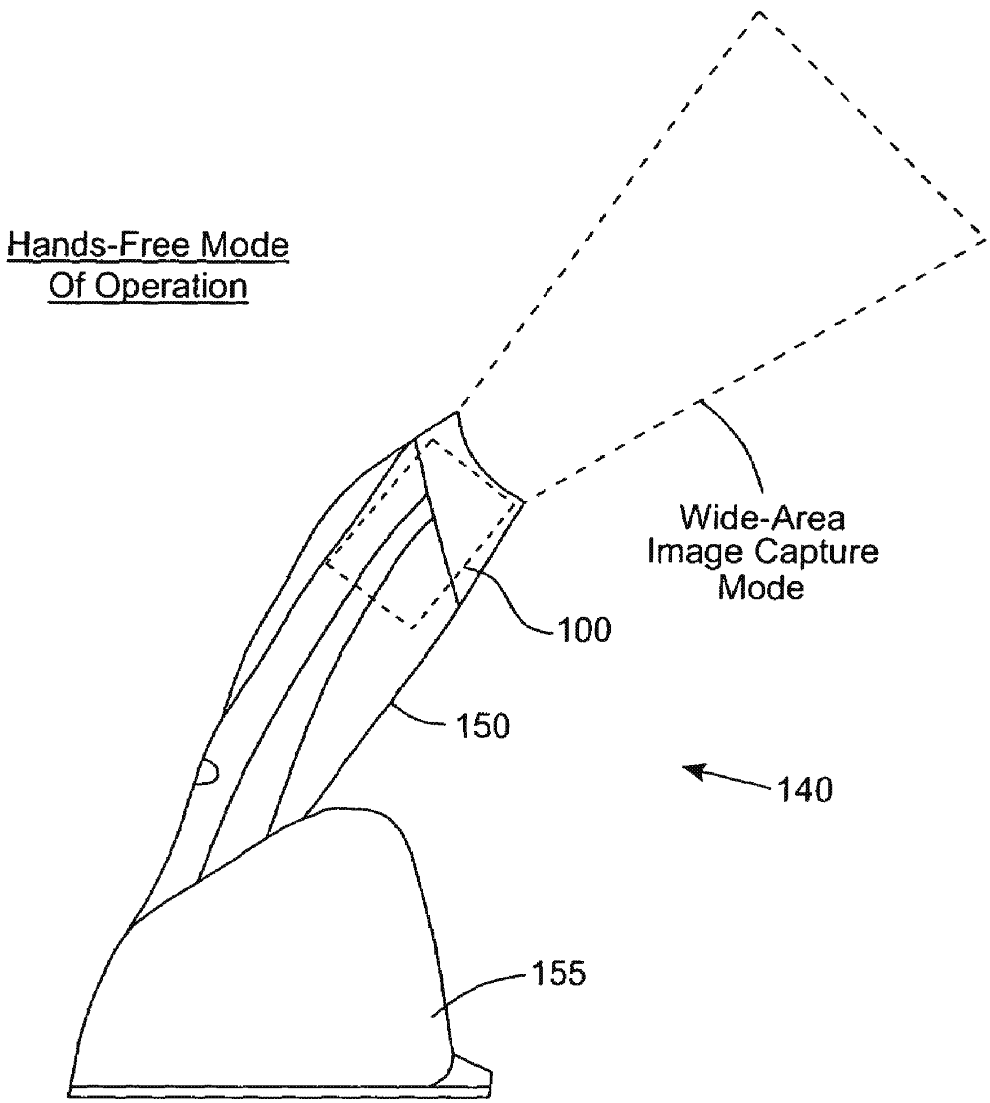
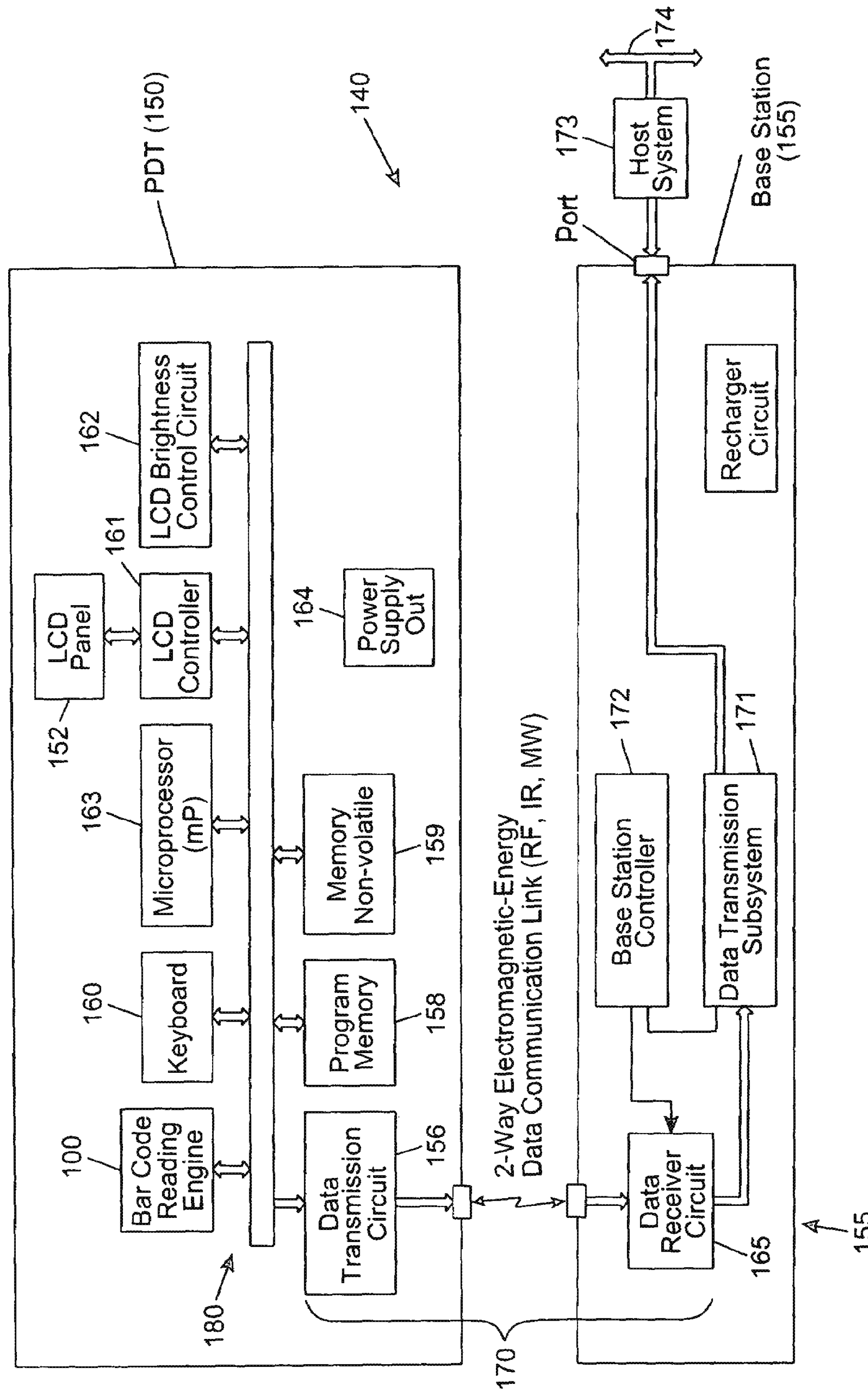

Hands-Free Mode
Of Operation
Wide-Area
Image Capture
Mode
100
150
140
155

FIG. 33



PDT (150)
140
174
173
Host System
Port
Base Station (155)
162
LCD Brightness Control Circuit
161
LCD Panel
LCD Controller
164
Power Supply Out
152
163
Microprocessor (mP)
Memory Non-volatile
159
160
Keyboard
Program Memory
158
100
Bar Code Reading Engine
Data Transmission Circuit
156
180
2-Way Electromagnetic-Energy Data Communication Link (RF, IR, MW)
170
172
Base Station Controller
171
Data Transmission Subsystem
Recharger Circuit
Data Receiver Circuit
165
155

FIG. 34

SYSTEM FOR DETECTING IMAGE LIGHT INTENSITY REFLECTED OFF AN OBJECT IN A DIGITAL IMAGING-BASED BAR CODE SYMBOL READING DEVICE

This application is a continuation-in-part of patent application Ser. No. 10/894,478 filed on Jul. 19, 2004, now U.S. Pat. No. 7,357,325, which is a continuation of patent application Ser. No. 10/712,787 filed on Nov. 13, 2003, now U.S. Pat. No. 7,128,266.

BACKGROUND OF INVENTION

1. Field of the Invention

The present invention relates to a method and system to ensure coincident camera and auto exposure optical axis in a digital imaging-based bar code symbol reading device for reading one-dimensional (1D) and two-dimensional (2D) bar code symbols, as well as other forms of graphically-encoded intelligence.

2. Brief Description of the State of Knowledge in the Art

The state of the automatic-identification industry can be understood in terms of (i) the different classes of bar code symbologies that have been developed and adopted by the industry, and (ii) the kinds of apparatus developed and used to read such bar code symbologies in various user environments.

In general, there are currently three major classes of bar code symbologies, namely: one dimensional (1D) bar code symbologies, such as UPC/EAN, Code 39, etc.; 1D stacked bar code symbologies, Code 49, PDF417, etc.; and two-dimensional (2D) data matrix symbologies.

One Dimensional optical bar code readers are well known in the art. Examples of such readers include readers of the Metrologic Voyager® Series Laser Scanner manufactured by Metrologic Instruments, Inc. Such readers include processing circuits that are able to read one dimensional (1D) linear bar code symbologies, such as the UPC/EAN code, Code 39, etc., that are widely used in supermarkets. Such 1D linear symbologies are characterized by data that is encoded along a single axis, in the widths of bars and spaces, so that such symbols can be read from a single scan along that axis, provided that the symbol is imaged with a sufficiently high resolution along that axis.

In order to allow the encoding of larger amounts of data in a single bar code symbol, a number of 1D stacked bar code symbologies have been developed, including Code 49, as described in U.S. Pat. No. 4,794,239 (Allais), and PDF417, as described in U.S. Pat. No. 5,340,786 (Pavlidis, et al.). Stacked symbols partition the encoded data into multiple rows, each including a respective 1D bar code pattern, all or most of all of which must be scanned and decoded, then linked together to form a complete message. Scanning still requires relatively high resolution in one dimension only, but multiple linear scans are needed to read the whole symbol.

The third class of bar code symbologies, known as 2D matrix symbologies offer orientation-free scanning and greater data densities and capacities than their 1D counterparts. In 2D matrix codes, data is encoded as dark or light data elements within a regular polygonal matrix, accompanied by graphical finder, orientation and reference structures. When scanning 2D matrix codes, the horizontal and vertical relationships of the data elements are recorded with about equal resolution.

In order to avoid having to use different types of optical readers to read these different types of bar code symbols, it is desirable to have an optical reader that is able to read symbols of any of these types, including their various subtypes, interchangeably and automatically. More particularly, it is desirable to have an optical reader that is able to read all three of the above-mentioned types of bar code symbols, without human intervention, i.e., automatically. This is turn, requires that the reader have the ability to automatically discriminate between and decode bar code symbols, based only on information read from the symbol itself. Readers that have this ability are referred to as "auto-discriminating" or having an "auto-discrimination" capability.

If an auto-discriminating reader is able to read only 1D bar code symbols (including their various subtypes), it may be said to have a 1D auto-discrimination capability. Similarly, if it is able to read only 2D bar code symbols, it may be said to have a 2D auto-discrimination capability. If it is able to read both 1D and 2D bar code symbols interchangeably, it may be said to have a 1D/2D auto-discrimination capability. Often, however, a reader is said to have a 1D/2D auto-discrimination capability even if it is unable to discriminate between and decode 1D stacked bar code symbols.

Optical readers that are capable of 1D auto-discrimination are well known in the art. An early example of such a reader is Metrologic's VoyagerCG® Laser Scanner, manufactured by Metrologic Instruments, Inc.

Optical readers, particularly hand held optical readers, that are capable of 1D/2D auto-discrimination and based on the use of an asynchronously moving 1D image sensor, are described in U.S. Pat. Nos. 5,288,985 and 5,354,977, which applications are hereby expressly incorporated herein by reference. Other examples of hand held readers of this type, based on the use of a stationary 2D image sensor, are described in U.S. Pat. Nos. 6,250,551; 5,932,862; 5,932,741; 5,942,741; 5,929,418; 5,914,476; 5,831,254; 5,825,006; 5,784,102, which are also hereby expressly incorporated herein by reference.

Optical readers, whether of the stationary or movable type, usually operate at a fixed scanning rate, which means that the readers are designed to complete some fixed number of scans during a given amount of time. This scanning rate generally has a value that is between 30 and 200 scans/sec for 1D readers. In such readers, the results the successive scans are decoded in the order of their occurrence.

Imaging-based bar code symbol readers have a number advantages over laser scanning based bar code symbol readers, namely: they are more capable of reading stacked 2D symbologies, such as the PDF417 symbology; more capable of reading matrix 2D symbologies, such as the Data Matrix symbology; more capable of reading bar codes regardless of their orientation; have lower manufacturing costs; and have the potential for use in other applications, which may or may not be related to bar code scanning, such as OCR, security systems, et.

Prior art imaging-based bar code symbol readers suffer from a number of additional shortcomings and drawbacks.

Most prior art hand held optical reading devices can be reprogrammed by reading bar codes from a bar code programming menu or with use of a local host processor as taught in U.S. Pat. No. 5,929,418. However, these devices are generally constrained to operate within the modes in which they have been programmed to operate, either in the field or on the bench, before deployment to end-user application environments. Consequently, the statically-configured nature of such prior art imaging-based bar code reading systems has limited their performance.

Prior art imaging-based bar code symbol readers with integrated illumination subsystems also support a relatively short

range of the optical depth of field. This limits the capabilities of such systems from reading big or highly dense bar code labels.

Prior art imaging-based bar code symbol readers generally require separate apparatus for producing a visible aiming beam to help the user to aim the camera's field of view at the bar code label on a particular target object.

Prior art imaging-based bar code symbol readers generally require capturing multiple frames of image data of a bar code symbol, and special apparatus for synchronizing the decoding process with the image capture process within such readers, as required in U.S. Pat. Nos. 5,932,862 and 5,942,741 assigned to Welch Allyn, Inc.

Prior art imaging-based bar code symbol readers generally require large arrays of LEDs in order to flood the field of view within which a bar code symbol might reside during image capture operations, oftentimes wasting larges amounts of electrical power which can be significant in portable or mobile imaging-based readers.

Prior art imaging-based bar code symbol readers generally require processing the entire pixel data set of capture images to find and decode bar code symbols represented therein.

Many prior art Imaging-Based Bar Code Symbol Readers require the use of decoding algorithms that seek to find the orientation of bar code elements in a captured image by finding and analyzing the code words of 2-D bar code symbologies represented therein.

Some prior art imaging-based bar code symbol readers generally require the use of a manually-actuated trigger to actuate the image capture and processing cycle thereof.

Prior art imaging-based bar code symbol readers generally require separate sources of illumination for producing visible aiming beams and for producing visible illumination beams used to flood the field of view of the bar code reader.

Prior art imaging-based bar code symbol readers generally utilize during a single image capture and processing cycle, and a single decoding methodology for decoding bar code symbols represented in captured images.

Some prior art imaging-based bar code symbol readers require exposure control circuitry integrated with the image detection array for measuring the light exposure levels on selected portions thereof.

Also, many imaging-based readers also require processing portions of captured images to detect the image intensities thereof and determine the reflected light levels at the image detection component of the system, and thereafter to control the LED-based illumination sources to achieve the desired image exposure levels at the image detector.

Prior art imaging-based bar code symbol readers employing integrated illumination mechanisms control image brightness and contrast by controlling the time the image sensing device is exposed to the light reflected from the imaged objects. While this method has been proven for the CCD-based bar code scanners, it is not suitable, however, for the CMOS-based image sensing devices, which require a more sophisticated shuttering mechanism, leading to increased complexity, less reliability and, ultimately, more expensive bar code scanning systems.

Prior art imaging-based bar code symbol readers generally require the use of tables and bar code menus to manage which decoding algorithms are to be used within any particular mode of system operation to be programmed by reading bar code symbols from a bar code menu.

Finally, as a result of limitations in the mechanical, electrical, optical, and software design of prior art imaging-based bar code symbol readers, such prior art readers generally (i) fail to enable users to read high-density 1D bar codes with the ease and simplicity of laser scanning based bar code symbol readers, and also 2D symbologies, such as PDF417 and Data Matrix, and (ii) are incapable of use in OCR and OCV, security applications, etc.

Thus, there is a great need in the art for an improved method of and apparatus for reading bar code symbols using image capture and processing techniques which avoid the shortcomings and drawbacks of prior art methods and apparatus.

OBJECTS AND SUMMARY OF THE PRESENT INVENTION

Accordingly, a primary object of the present invention is to provide a novel method of and system for enabling the reading of 1D and 2D bar code symbologies using image capture and processing based systems and devices, which avoid the shortcomings and drawbacks of prior art methods and apparatus.

Another object of the present invention is to provide a novel hand-supportable digital Imaging-Based Bar Code Symbol Reader capable of automatically reading 1D and 2D bar code symbologies using the state-of-the art imaging technology, and at the speed and with the reliability achieved by conventional laser scanning bar code symbol readers.

Another object of the present invention is to provide a novel hand-supportable digital Imaging-Based Bar Code Symbol Reader that is capable of reading stacked 2D symbologies such as PDF417, as well as Data Matrix.

Another object of the present invention is to provide a novel hand-supportable digital Imaging-Based Bar Code Symbol Reader that is capable of reading bar codes independent of their orientation with respect to the reader.

Another object of the present invention is to provide a novel hand-supportable digital Imaging-Based Bar Code Symbol Reader that is capable of reading high-density bar codes, as simply and effectively as "flying-spot" type laser scanners do.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reader capable of reading 1D and 2D bar code symbologies in a manner as convenient to the end users as when using a conventional laser scanning bar code symbol reader.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reader having an integrated LED-Based Multi-Mode Illumination Subsystem for generating a illumination beam for aiming on a target object and illuminating a 1D bar code symbol aligned therewith during an image capture mode of the system.

Another object of the present invention is to provide a method and system to ensure coincident camera and auto exposure optical axis in a digital imaging-based bar code symbol reading device.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reader employing a CMOS-type image sensing array using global exposure control techniques.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reader employing a continuously operating Automatic Light Exposure Measurement and Illumination Control Subsystem.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reader employing a Multi-Mode LED-Based Illumination Subsystem.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reading

System employing an integrated LED-Based Multi-Mode Illumination Subsystem driven by an Automatic Light Exposure Measurement and Illumination Control Subsystem responsive to control activation signals generated by a CMOS image sensing array and an IR-based Object Presence and Range Detection Subsystem during object illumination and image capturing operations.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reader employing a CMOS image sensing array which activates LED illumination driver circuitry to expose a target object to narrowly-tuned LED-based illumination when all of rows of pixels in said CMOS image sensing array are in a state of integration, thereby capturing high quality images independent of the relative motion between said bar code reader and the target object.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Reading System employing a mechanism of controlling the image brightness and contrast by controlling the time the illumination subsystem illuminates the target object, thus, avoiding the need for a complex shuttering mechanism for CMOS-based image sensing arrays employed therein.

Another object of the present invention is to provide a hand-supportable Imaging-Based Bar Code Symbol Reader having an integrated Multi-Mode Illumination Subsystem that supports an optical depth of field larger than conventional imaging-based bar code symbol readers.

These and other objects of the present invention will become more apparently understood hereinafter and in the claims to Invention appended hereto.

BRIEF DESCRIPTION OF THE DRAWINGS OF PRESENT INVENTION

For a more complete understanding of how to practice the Objects of the Present Invention, the following Detailed Description of the Illustrative Embodiments can be read in conjunction with the accompanying Drawings, briefly described below.

FIG. **1**A is a rear perspective view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**B is an front perspective view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**C is an elevated left side view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**D is an elevated right side view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**E is an elevated rear view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**F is an elevated front view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention, showing components associated with its illumination subsystem and its image capturing subsystem;

FIG. **1**G is a bottom view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**H is a top rear view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**I is a first perspective exploded view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**J is a second perspective exploded view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **1**K is a third perspective exploded view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention;

FIG. **2**A**1** is a schematic block diagram representative of a system design for the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device illustrated in FIGS. **1**A through **1**L, wherein the system design is shown comprising (1) a Multi-Mode Area-Type Image Formation and Detection (i.e. Camera) Subsystem having image formation (camera) optics for producing a field of view (FOV) upon an object to be imaged and a CMOS or like area-type image sensing array for detecting imaged light reflected off the object during illumination operations in either (i) a narrow-area image capture mode in which a few central rows of pixels on the image sensing array are enabled, or (ii) a wide-area image capture mode in which all rows of the image sensing array are enabled, (2) a Multi-Mode LED-Based Illumination Subsystem for producing narrow and wide area fields of narrow-band illumination within the FOV of the Image Formation And Detection Subsystem during narrow and wide area modes of image capture, respectively, so that only light transmitted from the Multi-Mode Illumination Subsystem and reflected from the illuminated object and transmitted through a narrow-band transmission-type optical filter realized within the hand-supportable housing (i.e. using a red-wavelength high-pass reflecting window filter element disposed at the light transmission aperture thereof and a low-pass filter before the image sensor) is detected by the image sensor and all other components of ambient light are substantially rejected, (3) an IR-based object presence and range detection subsystem for producing an IR-based object detection field within the FOV of the Image Formation and Detection Subsystem, (4) an Automatic Light Exposure Measurement and Illumination Control Subsystem for controlling the operation of the LED-Based Multi-Mode Illumination Subsystem, (5) an Image Capturing and Buffering Subsystem for capturing and buffering 2-D images detected by the Image Formation and Detection Subsystem, (6) a Multimode Image-Processing Based Bar Code Symbol Reading Subsystem for processing images captured and buffered by the Image Capturing and Buffering Subsystem and reading 1D and 2D bar code symbols represented, and (7) an Input/Output Subsystem for outputting processed image data and the like to an external host system or other information receiving or responding device, in which each said subsystem component is integrated about (8) a System Control Subsystem, as shown;

FIG. **2**A**2** is a schematic block representation of the multi-Mode Image-Processing Based Bar Code Symbol Reading Subsystem, realized using the three-tier computing platform illustrated in FIG. **2**B;

FIG. **2**B is schematic diagram representative of a system implementation for the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device illustrated in FIGS. **1**A through **2**A**2**, wherein the system implementation is shown comprising (1) an illumination board **33** carrying com-

ponents realizing electronic functions performed by the Multi-Mode LED-Based Illumination Subsystem and the Automatic Light Exposure Measurement And Illumination Control Subsystem, (2) a CMOS camera board carrying a high resolution (1280×1024 8-bit 6 micron pixel size) CMOS image sensor array running at 25 Mhz master clock, at 7 frames/second at 1280*1024 resolution with randomly accessible region of interest (ROI) window capabilities, realizing electronic functions performed by the multi-mode area-type Image Formation and Detection Subsystem, (3) a CPU board (i.e. computing platform) including (i) an Intel Sabinal 32-Bit Microprocessor PXA210 running at 200 Mhz 1.0 core voltage with a 16 bit 100 Mhz external bus speed, (ii) an expandable (e.g. 8+ megabyte) Intel J3 Asynchronous 16-bit Flash memory, (iii) an 16 Megabytes of 100 MHz SDRAM, (iv) an Xilinx Spartan II FPGA FIFO **39** running at 50 Mhz clock frequency and 60 MB/Sec data rate, configured to control the camera timings and drive an image acquisition process, (v) a multimedia card socket, for realizing the other subsystems of the system, (vi) a power management module for the MCU adjustable by the system bus, and (vii) a pair of UARTs (one for an IRDA port and one for a JTAG port), (4) an interface board for realizing the functions performed by the I/O subsystem, and (5) an IR-based object presence and range detection circuit for realizing the IR-based Object Presence And Range Detection Subsystem;

FIG. **3**A is a schematic representation showing the spatial relationships between the near and far and narrow and wide area fields of narrow-band illumination within the FOV of the Multi-Mode Image Formation and Detection Subsystem during narrow and wide area image capture modes of operation;

FIG. **3**B is a perspective partially cut-away view of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment, showing the LED-Based Multi-Mode Illumination Subsystem transmitting visible narrow-band illumination through its narrow-band transmission-type optical filter system and illuminating an object with such narrow-band illumination, and also showing the image formation optics, including the low pass filter before the image sensing array, for collecting and focusing light rays reflected from the illuminated object, so that an image of the object is formed and detected using only the optical components of light contained within the narrow-band of illumination, while all other components of ambient light are substantially rejected before image detection at the image sensing array;

FIG. **3**C is a schematic representation showing the geometrical layout of the optical components used within the hand-supportable Digital Imaging-Based Bar Code Reading Device of the first illustrative embodiment, wherein the red-wavelength reflecting high-pass lens element is positioned at the imaging window of the device before the image formation lens elements, while the low-pass filter is disposed before the image sensor of between the image formation elements, so as to image the object at the image sensing array using only optical components within the narrow-band of illumination, while rejecting all other components of ambient light;

FIG. **3**D is a schematic representation of the image formation optical subsystem employed within the hand-supportable Digital Imaging-Based Bar Code Reading Device of the first illustrative embodiment, wherein all three lenses are made as small as possible (with a maximum diameter of 12 mm), all have spherical surfaces, all are made from common glass, e.g. LAK2 (~LaK9), ZF10 (=SF8), LAF2 (~LaF3);

FIG. **3**E is a schematic representation of the lens holding assembly employed in the image formation optical subsystem of the hand-supportable Digital Imaging-Based Bar Code Reading Device of the first illustrative embodiment, showing a two-piece barrel structure which holds the lens elements, and a base structure which holds the image sensing array, wherein the assembly is configured so that the barrel structure slides within the base structure so as to focus the assembly;

FIG. **3F1** is a first schematic representation showing, from a side view, the physical position of the LEDs used in the Multi-Mode Illumination Subsystem, in relation to the image formation lens assembly, the image sensing array employed therein (e.g. a Motorola MCM20027 or National Semiconductor LM9638 CMOS 2-D image sensing array having a 1280×1024 pixel resolution (½" format), 6 micron pixel size, 13.5 Mhz clock rate, with randomly accessible region of interest (ROI) window capabilities);

FIG. **3F2** is a second schematic representation showing, from an axial view, the physical layout of the LEDs used in the Multi-Mode Illumination Subsystem of the Digital Imaging-Based Bar Code Reading Device, shown in relation to the image formation lens assembly, and the image sensing array employed therein;

FIG. **3**G is a flow chart describing the steps involved in determining the Depth of Field (DOF) of the image formation optics assembly employed in the bar code reading system of the present invention;

FIG. **4**A is a schematic representation of the Depth of Field Chart used in the design of the image formation optics in the Digital Imaging-Based Bar Code Reading Device, wherein image formation lens resolution characteristics are plotted against the pixel limits of the image sensing array;

FIG. **4**B is graphical chart illustrating the performance of the image formation optics of the Digital Imaging-Based Bar Code Reading Device of the present invention, plotting object distance (centimeters) against MTF values of image formation optics;

FIG. **4**C is a schematic representation illustrating the Depth of Field of the image formation optics of the Digital Imaging-Based Bar Code Reading Device of the present invention, measured in millimeters, and showing the narrowest bar code element dimension that can be measured over particular regions within its Depth of Field;

FIG. **4**D shows a DOF chart that plots the resolution of the image formation optics, indicating only the optical performance of the subsystem;

FIG. **4**E graphically illustrates how to read off the DOF for a certain mil size code, considering only the optical performance of the image formation optics of the Image Formation and Detection Subsystem;

FIG. **4**F shows the 1.4 and 1.6 pixel sampling limits plotted on the same axes as the optical performance curve for a fixed focal length reader (as they are functions of object distance);

FIG. **4**G graphically illustrates how to determine the composite DOF curve of the Image Formation and Detection Subsystem, considering optical performance and sampling limit together, for the 1.6 pixel case;

FIG. **4**H graphically illustrates how to read off the DOF for a certain mil size code, considering optical performance and sampling limit together, for the 1.6 pixel case;

FIGS. **4I1** through **4I3**, taken together, show an exemplary computer program written in ZPL (Zemax Programming Language) and capable of generating the composite DOF chart;

FIG. **5A1** is a schematic representation specifying the range of narrow-area illumination, near-field wide-area illumination, and far-field wide-area illumination produced from the LED-Based Multi-Mode Illumination Subsystem employed in the hand-supportable Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **5A2** is a table specifying the geometrical properties and characteristics of each illumination mode supported by the LED-Based Multi-Mode Illumination Subsystem employed in the hand-supportable Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **5B** is a schematic representation illustrating the physical arrangement of LED light sources associated with the narrow-area illumination array and the near-field and far-field wide-area illumination arrays employed in the Digital Imaging-Based Bar Code Reading Device of the present invention, wherein the LEDs in the far-field wide-area illuminating arrays are located behind spherical lenses, the LEDs in the narrow-area illuminating array are disposed behind cylindrical lenses, and the LEDs in the near-field wide-area illuminating array are unlensed in the first illustrative embodiment of the Digital Imaging-Based Bar Code Reading Device;

FIG. **5C1** is graphical representation showing the Lambertian emittance versus wavelength characteristics of the LEDs used to implement the narrow-area illumination array in the Multi-Mode Illumination Subsystem of the present invention;

FIG. **5C2** is graphical representation showing the Lambertian emittance versus polar angle characteristics of the LEDs used to implement the narrow-area illumination array in the Multi-Mode Illumination Subsystem of the present invention;

FIG. **5C3** is schematic representation of the cylindrical lenses used before the LEDs in the narrow-area (linear) illumination arrays in the Digital Imaging-Based Bar Code Reading Device of the present invention, wherein the first surface of the cylindrical lens is curved vertically to create a narrow-area (i.e. linear) illumination pattern, and the second surface of the cylindrical lens is curved horizontally to control the height of the of the narrow-area illumination pattern to produce a narrow-area (i.e. linear) illumination field;

FIG. **5C4** is a schematic representation of the layout of the pairs of LEDs and two cylindrical lenses used to implement the narrow-area (linear) illumination array employed in the Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **5C5** is a set of six illumination profiles for the narrow-area (linear) illumination fields produced by the narrow-area (linear) illumination array employed in the Digital Imaging-Based Bar Code Reading Device of the illustrative embodiment, taken at 30, 40, 50, 80, 120, and 220 millimeters along the field away from the imaging window (i.e. working distance) of the Digital Imaging-Based Bar Code Reading Device, illustrating that the spatial intensity of the narrow-area illumination field begins to become substantially uniform at about 80 millimeters;

FIG. **5D1** is graphical representation showing the Lambertian emittance versus wavelength characteristics of the LEDs used to implement the wide area illumination arrays employed in the Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **5D2** is graphical representation showing the Lambertian emittance versus polar angle characteristics of the LEDs used to implement the far-field and near-field wide-area illumination arrays employed in the Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **5D3** is schematic representation of the plano-convex lenses used before the LEDs in the far-field wide-area illumination arrays in the illumination subsystem of the present invention,

FIG. **5D4** is a schematic representation of the layout of LEDs and plano-convex lenses used to implement the far and narrow wide-area illumination array employed in the Digital Imaging-Based Bar Code Reading Device of the present invention, wherein the illumination beam produced therefrom is aimed by positioning the lenses at angles before the LEDs in the near-field (and far-field) wide-area illumination arrays employed therein;

FIG. **5D5** is a set of six illumination profiles for the near-field wide-area illumination fields produced by the near-field wide-area illumination arrays employed in the Digital Imaging-Based Bar Code Reading Device of the illustrative embodiment, taken at 10, 20, 30, 40, 60, and 100 millimeters along the field away from the imaging window (i.e. working distance) of the Digital Imaging-Based Bar Code Reading Device, illustrating that the spatial intensity of the near-field wide-area illumination field begins to become substantially uniform at about 40 millimeters;

FIG. **5D6** is a set of three illumination profiles for the far-field wide-area illumination fields produced by the far-field wide-area illumination arrays employed in the Digital Imaging-Based Bar Code Reading Device of the illustrative embodiment, taken at 100, 150 and 220 millimeters along the field away from the imaging window (i.e. working distance) of the Digital Imaging-Based Bar Code Reading Device, illustrating that the spatial intensity of the far-field wide-area illumination field begins to become substantially uniform at about 100 millimeters;

FIG. **5D7** is a table illustrating a preferred method of calculating the pixel intensity value for the center of the far-field wide-area illumination field produced from the Multi-Mode Illumination Subsystem employed in the Digital Imaging-Based Bar Code Reading Device of the present invention, showing a significant signal strength (greater than 80 DN);

FIG. **6A1** is a schematic representation showing the red-wavelength reflecting (high-pass) imaging window integrated within the hand-supportable housing of the Digital Imaging-Based Bar Code Reading Device, and the low-pass optical filter disposed before its CMOS image sensing array there within, cooperate to form a narrow-band optical filter subsystem for transmitting substantially only the very narrow band of wavelengths (e.g. 620-700 nanometers) of visible illumination produced from the Multi-Mode Illumination Subsystem employed in the Digital Imaging-Based Bar Code Reading Device, and rejecting all other optical wavelengths outside this narrow optical band however generated (i.e. ambient light sources);

FIG. **6A2** is schematic representation of transmission characteristics (energy versus wavelength) associated with the low-pass optical filter element disposed after the red-wavelength reflecting high-pass imaging window within the hand-supportable housing of the Digital Imaging-Based Bar Code Reading Device, but before its CMOS image sensing array, showing that optical wavelengths below 620 nanometers are transmitted and wavelengths above 620 nm are substantially blocked (e.g. absorbed or reflected);

FIG. **6A3** is schematic representation of transmission characteristics (energy versus wavelength) associated with the red-wavelength reflecting high-pass imaging window integrated within the hand-supportable housing of the Digital Imaging-Based Bar Code Reading Device of the present invention, showing that optical wavelengths above 700 nanometers are transmitted and wavelengths below 700 nm are substantially blocked (e.g. absorbed or reflected);

FIG. **6A4** is a schematic representation of the transmission characteristics of the narrow-based spectral filter subsystem integrated within the hand-supportable Imaging-Based Bar Code Symbol Reading Device of the present invention, plotted against the spectral characteristics of the LED-emissions

produced from the Multi-Mode Illumination Subsystem of the illustrative embodiment of the present invention;

FIG. **7**A is a schematic representation showing the geometrical layout of the spherical/parabolic light reflecting/collecting mirror and photodiode associated with the Automatic Light Exposure Measurement and Illumination Control Subsystem, and arranged within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the illustrative embodiment, wherein incident illumination is collected from a selected portion of the center of the FOV of the system using a spherical light collecting mirror, and then focused upon a photodiode for detection of the intensity of reflected illumination and subsequent processing by the Automatic Light Exposure Measurement and Illumination Control Subsystem, so as to then control the illumination produced by the LED-based Multi-Mode Illumination Subsystem employed in the Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **7A1** is a schematic representation showing the geometrical layout of the mirrored beam splitter and photodiode associated with the Automatic Light Exposure Measurement and Illumination Control Subsystem, and arranged within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the illustrative embodiment, wherein illumination is directed from the center of the FOV of the system to a mirrored beam splitter, and a portion of the illumination is transmitted through the beam splitter and focused upon a photodiode for detection of the intensity of reflected illumination and subsequent processing by the Automatic Light Exposure Measurement and Illumination Control Subsystem, so as to then control the illumination produced by the LED-based Multi-Mode Illumination Subsystem employed in the Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **7A2** is a schematic representation showing the geometrical layout of the cube-type beam splitter and photodiode associated with the Automatic Light Exposure Measurement and Illumination Control Subsystem, and arranged within the hand-supportable Digital imaging-Based Bar Code Symbol Reading Device of the illustrative embodiment, wherein illumination is directed from the center of the FOV of the system to a cube-type beam splitter, and a portion of the illumination is transmitted through the beam splitter and focused upon a photodiode for detection of the intensity of reflected illumination and subsequent processing by the Automatic Light Exposure Measurement and Illumination Control Subsystem, so as to then control the illumination produced by the LED-based Multi-Mode Illumination Subsystem employed in the Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **7**B is a schematic diagram of the Automatic Light Exposure Measurement and Illumination Control Subsystem employed in the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, wherein illumination is collected from the center of the FOV of the system and automatically detected so as to generate a control signal for driving, at the proper intensity, the narrow-area illumination array as well as the far-field and narrow-field wide-area illumination arrays of the Multi-Mode Illumination Subsystem, so that the CMOS image sensing array produces digital images of illuminated objects of sufficient brightness;

FIG. **7**C is a schematic diagram of a hybrid analog/digital circuit designed to implement the Automatic Light Exposure Measurement and Illumination Control Subsystem of FIG. **7**B employed in the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention;

FIG. **7**D is a schematic diagram showing that, in accordance with the principles of the present invention, the CMOS image sensing array employed in the Digital Imaging-Based Bar Code Reading Device of the illustrative embodiment, once activated by the System Control Subsystem (or directly by the trigger switch), and when all rows in the image sensing array are in a state of integration operation, automatically activates the Automatic Light Exposure Measurement and Illumination Control Subsystem which, in response thereto, automatically activates the LED illumination driver circuitry to automatically drive the appropriate LED illumination arrays associated with the Multi-Mode Illumination Subsystem in a precise manner and globally expose the entire CMOS image detection array with narrowly tuned LED-based illumination when all of its rows of pixels are in a state of integration, and thus have a common integration time, thereby capturing high quality images independent of the relative motion between the bar code reader and the object;

FIGS. **7E1** and **7**E**2**, taken together, set forth a flow chart describing the steps involved in carrying out the global exposure control method of the present invention, within the Digital Imaging-Based Bar Code Reading Device of the illustrative embodiment;

FIG. **8** is a schematic block diagram of the IR-based automatic Object Presence and Range Detection Subsystem employed in the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, wherein a first range indication control signal is generated upon detection of an object within the near-field region of the Multi-Mode Illumination Subsystem, and wherein a second range indication control signal is generated upon detection of an object within the far-field region of the Multi-Mode Illumination Subsystem;

FIG. **9** is a schematic representation of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, showing that its CMOS image sensing array is operably connected to its microprocessor through a FIFO (realized by way of a FPGA) and a system bus, and that its SDRAM is also operably connected to the microprocessor by way of the system bus, enabling the mapping of pixel data captured by the imaging array into the SDRAM under the control of the direct memory access (DMA) module within the microprocessor;

FIG. **10** is a schematic representation showing how the bytes of pixel data captured by the CMOS imaging array within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, are mapped into the addressable memory storage locations of its SDRAM during each image capture cycle carried out within the device;

FIG. **11** is a schematic representation showing the software modules associated with the three-tier software architecture of the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, namely: the Main Task module, the CodeGate Task module, the Metroset Task module, the Application Events Manager module, the User Commands Table module, and the Command Handler module residing with the Application layer of the software architecture; the Tasks Manager module, the Events Dispatcher module, the Input/Output Manager module, the User Commands Manager module, the Timer Subsystem module, the Input/Output Subsystem module and the Memory Control Subsystem module residing with the System Core (SCORE) layer of the software architecture; and the Linux Kemal module, the Linux File System module, and Device Drivers modules residing within the Linux Operating System (OS) layer of the software architecture;

FIG. **12**A is a schematic representation of the Events Dispatcher software module which provides a means of signaling and delivering events to the Application Events Manager, including the starting of a new task, stopping a currently running task, doing something, or doing nothing and ignoring the event;

FIG. **12**B is a Table listing examples of System-Defined Events which can occur and be dispatched within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, namely: SCORE_EVENT_POWER_UP which signals the completion of system start-up and involves no parameters; _SCORE_EVENT_TIMEOUT which signals the timeout of the logical timer, and involves the parameter "pointer to timer id"; SCORE_EVENT_UNEXPECTED_INPUT which signals that the unexpected input data is available and involves the parameter "pointer to connection id"; SCORE_EVENT_ TRIG_ON which signals that the user pulled the trigger switch and involves no parameters; SCORE_EVENT_ TRIG_OFF which signals that the user released the trigger switch and involves no parameters; SCORE_EVENT_OBJECT_DETECT_ON which signals that the object is positioned under the bar code reader and involves no parameters; SCORE_EVENT_OBJECT_DETECT_OFF which signals that the object is removed from the field of view of the bar code reader and involves no parameters; SCORE_EVENT_ EXIT_TASK which signals the end of the task execution and involves the pointer UTID; and SCORE_EVENT_ ABORT_ TASK which signals the aborting of a task during execution;

FIG. **12**C is a schematic representation of the Tasks Manager software module which provides a means for executing and stopping application specific tasks (i.e. threads);

FIG. **12**D is a schematic representation of the Input/Output Manager software module (i.e. Input/Output Subsystem), which runs in the background and monitors activities of external devices and user connections, and signals appropriate events to the Application Layer, which such activities are detected;

FIGS. **12**E**1** and **12**E**2** set forth a schematic representation of the Input/Output Subsystem software module which provides a means for creating and deleting input/output connections, and communicating with external systems and devices;

FIGS. **12**F**1** and **12**F**2** set forth a schematic representation of the Timer Subsystem which provides a means for creating, deleting, and utilizing logical timers;

FIGS. **12**G**1** and **12**G**2** set forth a schematic representation of the Memory Control Subsystem which provides an interface for managing the thread-level dynamic memory with the device, fully compatible with standard dynamic memory management functions, as well as a means for buffering collected data;

FIG. **12**H is a schematic representation of the User Commands Manager which provides a standard way of entering user commands, and executing application modules responsible for handling the same;

FIG. **12**I is a schematic representation of the Device Driver software modules, which includes trigger switch drivers for establishing a software connection with the hardware-based manually-actuated trigger switch employed on the Digital Imaging-Based Bar Code Reading Device, an image acquisition driver for implementing image acquisition functionality aboard the Digital Imaging-Based Bar Code Reading Device, and an IR driver for implementing object detection functionality aboard the Imaging-Based Bar Code Symbol Reading Device;

FIG. **13**A is an exemplary flow chart representation showing how when the user points the bar code reader towards a bar code symbol, the IR device drivers detect that object within the field, and then wakes up the Input/Output Manager software module at the System Core Layer;

FIG. **13**B is an exemplary flow chart representation showing how upon detecting an object, the Input/Output Manager posts the SCORE_OBJECT_DETECT_ON event to the Events Dispatcher software module;

FIG. **13**C is an exemplary flow chart representation showing how, in response to detecting an object, the Events Dispatcher software module passes the SCORE_OBJECT_DETECT_ON event to the Application Layer;

FIG. **13**D is an exemplary flow chart representation showing how upon receiving the SCORE_OBJECT_DETECT_ON event at the Application Layer, the Application Events Manager executes an event handling routine which activates the narrow-area illumination array associated with the Multi-Mode Illumination Subsystem, and executes the CodeGate Task described in FIG. **13**E;

FIG. **13**E is an exemplary flow chart representation showing how what operations are carried out when the CodeGate Task is executed within the Application Layer;

FIG. **13**F is an exemplary flow chart representation showing how, when the user pulls the trigger switch on the bar code reader while the Code Task is executing, the trigger device driver wakes up the Input/Output Manager at the System Core Layer;

FIG. **13**G is an exemplary flow chart representation showing how, in response to waking up, the Input/Output Manager posts the SCORE_TRIGGER_ON event to the Events Dispatcher;

FIG. **13**H is an exemplary flow chart representation showing how the Events Dispatcher passes on the SCORE_TRIGGER_ON event to the Application Events Manager at the Application Layer;

FIG. **13**I is an exemplary flow chart representation showing how the Application Events Manager responds to the SCORE_TRIGGER_ON event by invoking a handling routine within the Task Manager at the System Core Layer which deactivates the narrow-area illumination array associated with the Multi-Mode Illumination Subsystem, cancels the CodeGate Task, and executes the Main Task;

FIG. **13**J is an exemplary flow chart representation showing what operations are carried out when the Main Task is executed within the Application Layer;

FIG. **13**K is an exemplary flow chart representation showing what operations are carried out when the Data Output Procedure, called in the Main Task, is executed within the Input/Output Subsystem software module in the Application Layer;

FIG. **13**L is an exemplary flow chart representation showing decoded symbol character data being sent from the Input/Output Subsystem to the Device Drivers within the Linux OS Layer of the system;

FIG. **13**M is a flow chart describing a novel method of generating wide-area illumination, for use during the Main Task routine so as to illuminate objects with a wide-area illumination field in a manner, which substantially reduces specular-type reflection at the CMOS image sensing array in the Digital Imaging-Based Bar Code Reading Device of the present invention;

FIG. **14** is a table listing various bar code symbologies supported by the Multi-Mode Bar Code Symbol Reading Subsystem module employed within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention;

FIG. **15** is a table listing the four primary modes in which the Multi-Mode Bar Code Symbol Reading Subsystem module can be programmed to operate, namely: the Automatic Mode wherein the Multi-Mode Bar Code Symbol Reading Subsystem is configured to automatically process a captured frame of digital image data so as to search for one or more bar codes represented therein in an incremental manner, and to continue searching until the entire image is processed; the Manual Mode wherein the Multi-Mode Bar Code Symbol Reading Subsystem is configured to automatically process a captured frame of digital image data, starting from the center or sweep spot of the image at which the user would have aimed the bar code reader, so as to search for (i.e. find) one or more bar code symbols represented therein, by searching in a helical manner through frames or blocks of extracted image feature data and marking the same and processing the corresponding raw digital image data until a bar code symbol is recognized/read within the captured frame of) image data; the ROI-Specific Mode wherein the Multi-Mode Bar Code Symbol Reading Subsystem is configured to automatically process a specified "region of interest" (ROI) in a captured frame of digital image data so as to search for one or more bar codes represented therein, in response to coordinate data specifying the location of the bar code within the field of view of the multi-mode image formation and detection system; the NoFinder Mode wherein the Multi-Mode Bar Code Symbol Reading Subsystem is configured to automatically process a captured narrow-area (linear) frame of digital image data, without feature extraction and marking operations used in the Automatic and Manual Modes, so as read one or more bar code symbols represented therein; and the Omniscan Mode, wherein the Multi-Mode Bar Code Symbol Reading Subsystem is configured to automatically process a captured frame of digital image data along any one or more predetermined virtual scan line orientations, without feature extraction and marking operations used in the Automatic and Manual Modes, so as to read one or more bar code symbols represented therein;

FIG. **16** is a exemplary flow chart representation showing the steps involved in setting up and cleaning up the software sub-Application entitled "Multi-Mode Image-Processing Based Bar Code Symbol Reading Subsystem", once called from either (i) the CodeGate Task software module at the Block entitled READ BAR CODE(S) IN CAPTURED NARROW-AREA IMAGE indicated in FIG. **13**E, or (ii) the Main Task software module at the Block entitled "READ BAR CODE(S) IN CAPTURED WIDE-AREA IMAGE" indicated in FIG. **13**J;

FIG. **17**A is a summary of the steps involved in the decode process carrying out by the Multi-Mode Bar Code Symbol Reading Subsystem of the present invention during its Automatic Mode of operation, wherein (1) the first stage of processing involves searching for (i.e. finding) regions of interest (ROIs) by processing a low resolution image of a captured frame of high-resolution image data, partitioning the low-resolution image into N×N blocks, and creating a feature vector for each block using spatial-derivative based image processing techniques, (2) the second stage of processing involves marking ROIs by examining the feature vectors for regions of high-modulation, calculating bar code orientation and marking the four corners of a bar code as a ROI, and (3) the third stage of processing involves reading any bar code symbols represented within the ROI by traversing the bar code and updating the feature vectors, examining the zero-crossings of filtered images, creating bar and space patterns, and decoding the bar and space patterns using conventional decoding algorithms;

FIG. **17**B is an exemplary flow chart representation of the steps involved in the image-processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its Automatic Mode of operation;

FIG. **18**A is a graphical representation illustrating the generation of a low-resolution image of a package label from an original high-resolution image thereof during the first finding stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem configured in its Automatic Mode of operation;

FIG. **18**B is a graphical representation illustrating the partitioning of the low-resolution image of the package label, the calculation of feature vectors using the same, and the analysis of these feature vectors for parallel lines, during the first finding stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem during its Automatic Mode of operation;

FIG. **18**C is a graphical representation showing that the calculation of feature vectors within each block of low-resolution image data, during the second marking stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem, can involve the use of gradient vectors, edge density measures, the number of parallel edge vectors, centroids of edges, intensity variance, and the histogram of intensities captured from the low-resolution image;

FIG. **18**D is a graphical representation of the examination of feature vectors looking for high edge density, large number of parallel edge vectors and large intensity variance, during the second marking stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem during its Automatic Mode of operation;

FIGS. **18**E and **18**F set forth graphical representations of calculating bar code orientation during the second marking stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem operating in its Automatic Mode, wherein each feature vector block, the bar code is traversed (i.e. sliced) at different angles, the slices are matched with each other based on "least mean square error", and the correct orientation is determined to be that angle which matches the mean square error sense through every slice of the bar code symbol represented within the captured image;

FIG. **18**F is a graphical representation of calculating bar code orientation, during the second marking stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem operating in its Automatic Mode;

FIG. **18**G is a graphical representation of the marking of the four corners of the detected bar code symbol during the second marking stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem operating in its Automatic Mode, wherein such marking operations are performed on the full high-resolution image of the parcel, the bar code is traversed in either direction starting from the center of the block, the extent of modulation is detected using the intensity variance, and the x,y coordinates (pixels) of the four corners of the bar code are detected starting from 1 and 2 and moving perpendicular to the bar code orientation, and define the ROI by the detected four corners of the bar code symbol within the high-resolution image;

FIG. **18**H is a graphical representation of updating the feature vectors during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem operating in its Automatic Mode, wherein the histogram component of the feature vector Fv is updated while traversing the bar code symbol, the estimate of the black-to-white transition is calculated, and an estimate of narrow and wide elements of the bar code symbol are calculated;

FIG. **18**I is a graphical representation of the search for zero crossings during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem operating in its Automatic Mode, wherein the high-resolution bar code image is median filtered in a direction perpendicular to bar code orientation, the second derivative zero crossings define edge crossings, the zero-crossing data is used only for detecting edge transitions, and the black/white transition estimates are used to put upper and lower bounds on the grey levels of the bars and spaces of the bar code symbol represented within the captured image;

FIG. **18**J is a graphical representation of creating bar and space pattern during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem operating in its Automatic Mode, wherein the edge transition is modeled as a ramp function, the edge transition is assumed to be 1 pixel wide, the edge transition location is determined at the subpixel level, and the bar and space counts are gathered using edge transition data;

FIG. **18**K is a graphical representation of the decode bar and space pattern during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem operating in its Automatic Mode, wherein the bar and space data is framed with borders, and the bar and space data is decoded using existing laser scanning bar code decoding algorithms;

FIG. **19**A is a summary of the steps involved in the image-processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its Manual Mode of operation, wherein (1) the first stage of processing involves searching for (i.e. finding) regions of interest (ROIs) by processing a low resolution image of a captured frame of high-resolution image data, partitioning the low-resolution image into N×N blocks, and creating a feature vector for the middle block using spatial-derivative based image processing techniques, (2) the second stage of processing involves marking ROIs by examining the feature vectors for regions of high-modulation and returning to the first stage to create feature vectors for other blocks surrounding the middle block (in a helical manner), calculating bar code orientation and marking the four corners of a bar code as a ROI, and (3) the third stage of processing involves reading any bar code symbols represented within the ROI by traversing the bar code and updating the feature vectors, examining the zero-crossings of filtered images, creating bar and space patterns, and decoding the bar and space patterns using conventional decoding algorithms;

FIG. **19**B is an exemplary flow chart representation of the steps involved in the image-processing method carrying out by the Multi-Mode Bar Code Symbol Reading Subsystem during its Manual Mode of operation;

FIG. **20**A is a summary of the steps involved in the image processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its NoFinder Mode of operation, wherein the Decoder Module does not employ bar code element finding or marking techniques (i.e. Finder Module and Marker Module) and directly processes a narrow-area portion of a captured high-resolution image, starting from the middle thereof, examines the zero-crossings of the filtered image, creates bar and space patterns therefrom, and then decodes the bar and space patterns using conventional decoding algorithms;

FIG. **20**B is an exemplary flow chart representation of the steps involved in the image-processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its NoFinder Mode of operation;

FIG. **21**A is a summary of the steps involved in the image-processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its OmniScan Mode of operation, wherein the Decoder Module does not employ bar code element finding or marking techniques (i.e. Finder Module and Marker Module), assumes the imaged bar code symbol resides at the center of the captured wide-area high-resolution image with about a 1:1 aspect ratio, and directly processes the high-resolution image along a set of parallel spaced-apart (e.g. 50 pixels) virtual scan lines, examines the zero-crossings along the virtual scan lines, creates bar and space patterns therefrom, and then decodes the bar and space patterns, with the option of reprocessing the high-resolution image along a different set of parallel spaced-apart virtual scan lines oriented at a different angle from the previously processed set of virtual scan lines (e.g. 0, 30, 60, 90, 120 or 150 degrees);

FIG. **21**B is an exemplary flow chart representation of the steps involved in the image-processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its OmniScan Mode of operation;

FIG. **22**A is a summary of the steps involved in the image-processing based bar code reading method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem of the present invention during its "ROI-Specific" Mode of operation, designed for use in combination with the Omniscan Mode of operation, wherein (1) the first stage of processing involves receiving region of interest (ROI) coordinates (x1, x2) obtained during the Omniscan Mode of operation (after the occurrence of a failure to decode), re-partitioning the captured low-resolution image (from the Omniscan Mode) into N×N blocks, and creating a feature vector for the ROI-specified block(s) using spatial-derivative based image processing techniques, (2) the second stage of processing involves marking additional ROIs by examining the feature vectors for regions of high-modulation and returning to the first stage to create feature vectors for other blocks surrounding the middle block (in a helical manner), calculating bar code orientation and marking the four corners of a bar code as a ROI, and (3) the third stage of processing involves reading any bar code symbols represented within the ROI by traversing the bar code symbol and updating the feature vectors, examining the zero-crossings of filtered images, creating bar and space patterns, and decoding the bar and space patterns using conventional decoding algorithms;

FIG. **22**B is an exemplary flow chart representation of the steps involved in the image-processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem of the present invention during its ROI-specific Mode of operation;

FIG. **23** is a specification of Multi-Mode Bar Code Symbol Reading Subsystem operated during its first multi-read (Omniscan/ROI-Specific) mode of operation;

FIG. **24** is a specification of Multi-Mode Bar Code Symbol Reading Subsystem operated during its second multi-read (No-Finder/ROI-Specific) mode of operation;

FIG. **25** is a specification of Multi-Mode Bar Code Symbol Reading Subsystem operated during its third multi-read (No-Finder/Omniscan/ROI-Specific) mode of operation; and

FIGS. **26**A; and **26**B taken together, provide a table listing the primary Programmable Modes of Bar Code Reading Operation within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, namely: Programmed Mode of System Operation No. 1—Manually-Triggered Single-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode Of System Operation No. 2—Manually-Triggered Multiple-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode Of System Operation No. 3—Manually-

Triggered Single-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode And The Automatic Or Manual Modes of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 4—Manually-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode And The Automatic Or Manual Modes of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 5—Manually-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing the No-Finder Mode And The Automatic Or Manual Modes of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 6—Automatically-Triggered Single-Attempt 1D Single-Read Mode Employing The No-Finder Mode Of the Multi-Mode Bar Code Reading Subsystem: Programmed Mode of System Operation No. 7—Automatically-Triggered Multi-Attempt iD Single-Read Mode Employing The No-Finder Mode Of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 8—Automatically-Triggered Multi-Attempt 1D/2D Single-Read Mode Employing The No-Finder Mode and Manual and/or Automatic Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 9—Automatically-Triggered Multi-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder Mode and Manual and/or Automatic Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of System Operation No. 10—Automatically-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing The Manual, Automatic or Omniscan Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 11—Semi-Automatic-Triggered Single-Attempt 1D/2D Single-Read Mode Employing The No-Finder Mode And The Automatic Or Manual Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of System Operation No. 12—Semi-Automatic-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing The No-Finder Mode And The Automatic Or Manual Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of Operation No. 13—Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder Mode And The Automatic Or Manual Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of Operation No. 14—Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder Mode And The Omniscan Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of Operation No. 15Continuously—Automatically-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The Automatic, Manual Or Omniscan Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of System Operation No. 16—Diagnostic Mode Of Imaging-Based Bar Code Reader Operation:; and Programmable Mode of System Operation No. 17—Live Video Mode Of Imaging-Based Bar Code Reader Operation;

FIG. **27**A is a schematic representation specifying the four modes of illumination produced from the Multi-Mode Illumination Subsystem employed in the second illustrative embodiment of the Digital Imaging-Based Bar Code Symbol Reader of the present invention, which supports both near and far fields of narrow-area illumination generated during the narrow-area image capture mode of its Multi-Mode Image Formation and Detection Subsystem;

FIG. **27**B is a schematic representation specifying how the cylindrical beam shaping optics employed within near-field and far-field narrow-area illumination arrays can be easily tailored to generate near and far narrow-area illumination fields having geometrical characteristics that enables (i) simple reading of extended-length bar code symbols within the far-field region of the FOV of the system, and also (ii) simple reading of bar code menus with a great degree of control within the near-field region of the FOV, preferably during a “Semi-Automatic-Triggered” programmed mode of system operation;

FIG. **28** is a schematic representation illustrating the physical arrangement of LEDs and light focusing lenses associated with the near and far field narrow-area and wide-area illumination arrays employed in the Digital Imaging-Based Bar Code Symbol Reading Device according to the second illustrative embodiment of the present invention;

FIG. **29**A is a first perspective view of a second illustrative embodiment of the portable POS Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, shown having a hand-supportable housing of a different form factor than that of the first illustrative embodiment, and configured for use in its hands-free/presentation mode of operation, supporting primarily wide-area image capture;

FIG. **29**B is a second perspective view of the second illustrative embodiment of the portable POS Digital Imaging-Based Bar Code Reading Device of the present invention, shown configured and operated in its hands-free/presentation mode of operation, supporting primarily wide-area image capture;

FIG. **29**C is a third perspective view of the second illustrative embodiment of the portable Digital Imaging-Based Bar Code Reading Device of the present invention, showing configured and operated in a hands-on type mode, supporting both narrow and wide area modes of image capture;

FIG. **30** is a perspective view of a third illustrative embodiment of the Digital Imaging-Based Bar Code Symbol Reading Device of the present invention, realized in the form of a Multi-Mode Image Capture And Processing Engine that can be readily integrated into various kinds of information collection and processing systems, including wireless portable data terminals (PDTs), reverse-vending machines, retail product information kiosks and the like;

FIG. **31** is a schematic representation of a Wireless Bar Code-Driven Portable Data Terminal embodying the Imaging-Based Bar Code Symbol Reading Engine of the present invention, shown configured and operated in a hands-on mode;

FIG. **32** is a perspective view of the Wireless Bar Code Driven Portable Data Terminal of FIG. **31** shown configured and operated in a hands-on mode, wherein the Imaging-Based Bar Code Symbol Reading Engine embodied therein is used to read a bar code symbol on a package and the symbol character data representative of the read bar code is being automatically transmitted to its cradle-providing base station by way of an RF-enabled 2-way data communication link;

FIG. **33** is a side view of the Wireless Bar Code Driven Portable Data Terminal of FIGS. **31** and **32** shown configured and operated in a hands-free mode, wherein the Imaging-Based Bar Code Symbol Reading Engine is configured in a wide-area image capture mode of operation, suitable for presentation-type bar code reading at point of sale (POS) environments; and

FIG. **34** is a block schematic diagram showing the various subsystem blocks associated with a design model for the Wireless Hand-Supportable Bar Code Driven Portable Data Terminal System of FIGS. **31**, **32** and **33**, shown interfaced with possible host systems and/or networks.

For a more complete understanding of how to practice the Objects of the Present Invention, the following Detailed

Description of the Illustrative Embodiments can be read in conjunction with the accompanying Drawings, briefly described below.

DETAILED DESCRIPTION OF THE ILLUSTRATIVE EMBODIMENTS OF THE PRESENT INVENTION

Referring to the figures in the accompanying Drawings, the various illustrative embodiments of the hand-supportable imaging-based bar code symbol reading system of the present invention will be described in great detail, wherein like elements will be indicated using like reference numerals.

Hand-Supportable Digital Imaging-Based Bar Code Reading Device of the First Illustrative Embodiment of the Present Invention

Referring to FIGS. **1**A through **1**K, the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the first illustrative embodiment of the present invention **1** is shown in detail comprising a hand-supportable housing **2** having a handle portion **2**A and a head portion **2**B that is provided with a light transmission window **3** with a high-pass (red-wavelength reflecting) optical filter element **4**A having light transmission characteristics set forth in FIG. **6**A**2**, in the illustrative embodiment. As will be described in greater detail hereinafter, high-pass optical filter element **4**A cooperates within an interiorly mounted low-pass optical filter element **4**B characterized in FIG. **6**A**1**, which cooperates with the high-pass optical filter element **4**A. These high and low pass filter elements **4**A and **4**B cooperate to provide a narrow-band optical filter system **4** that integrates with the head portion of the housing and permits only a narrow band of illumination (e.g. 633 nanometers) to exit and enter the housing during imaging operations.

As best shown in FIGS. **1**I, **1**J, and **1**K, the hand-supportable housing **2** of the illustrative embodiment comprises: left and right housing handle halves **2**A**1** and **2**A**2**; a foot-like structure **2**A**3** which is mounted between the handle halves **2**A**1** and **2**A**2**; a trigger switch structure **2**C which snap fits within and pivots within a pair of spaced apart apertures **2**D**1** and **2**D**2** provided in the housing halves; a light transmission window panel **5** through which light transmission window **3** is formed and supported within a recess formed by handle halves **2**A**1** and **2**A**2** when they are brought together, and which supports all LED illumination arrays provided by the system; an optical bench **6** for supporting electro-optical components and operably connected an orthogonally-mounted PC board **7** which is mounted within the handle housing halves; a top housing portion **2**B**1** for connection with the housing handle halves **2**A**1** and **2**A**2** and enclosing the head portion of the housing; light pipe lens element **8** for mounting over an array of light emitting diodes (LEDs) **9** and light pipe structures **10** mounted within the rear end of the head portion of the hand-supportable housing; and a front bumper structure **2**E for holding together the top housing portion **2**B**1** and left and right handle halves **2**A**1** and **2**A**2** with the light transmission window panel **5** sandwiched there between, while providing a level of shock protection thereto.

In other embodiments of the present invention shown in FIGS. **27** through **33** the form factor of the hand-supportable housing might be different. In yet other applications, the housing need not even be hand-supportable, but rather might be designed for stationary support on a desktop or countertop surface, or for use in a commercial or industrial application.

Schematic Block Functional Diagram as System Design Model for the Hand-Supportable Digital Image-Based Bar Code Reading Device of the Present Invention

As shown in the system design model of FIG. **2**A**1**, the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device **1** of the illustrative embodiment comprises: an IR-based Object Presence and Range Detection Subsystem **12**; a Multi-Mode Area-type Image Formation and Detection (i.e. camera) Subsystem **13** having narrow-area mode of image capture, near-field wide-area mode of image capture, and a far-field wide-area mode of image capture; a Multi-Mode LED-Based Illumination Subsystem **14** having narrow-area mode of illumination, near-field wide-area mode of illumination, and a far-field wide-area mode of illumination; an Automatic Light Exposure Measurement and Illumination Control Subsystem **15**; an Image Capturing and Buffering Subsystem **16**; a Multi-Mode Image-Processing Bar Code Symbol Reading Subsystem **17** having five modes of image-processing based bar code symbol reading indicated in FIG. **2**A**2** and to be described in detail hereinabove; an Input/Output Subsystem **18**; a manually-actuatable trigger switch **2**C for sending user-originated control activation signals to the device; a System Mode Configuration Parameter Table **70**; and a System Control Subsystem **18** integrated with each of the above-described subsystems, as shown.

The primary function of the IR-based Object Presence and Range Detection Subsystem **12** is to automatically produce an IR-based object detection field **20** within the FOV of the Multi-Mode Image Formation and Detection Subsystem **13**, detect the presence of an object within predetermined regions of the object detection field (**20**A, **20**B), and generate control activation signals A**1** which are supplied to the System Control Subsystem **19** for indicating when and where an object is detected within the object detection field of the system.

In the first illustrative embodiment, the Multi-Mode Image Formation And Detection (I.E. Camera) Subsystem **13** has image formation (camera) optics **21** for producing a field of view (FOV) **23** upon an object to be imaged and a CMOS area-image sensing array **22** for detecting imaged light reflected off the object during illumination and image acquisition/capture operations.

In the first illustrative embodiment, the primary function of the Multi-Mode LED-Based Illumination Subsystem **14** is to produce a narrow-area illumination field **24**, near-field wide-area illumination field **25**, and a far-field wide-area illumination field **25**, each having a narrow optical-bandwidth and confined within the FOV of the Multi-Mode Image Formation And Detection Subsystem **13** during narrow-area and wide-area modes of imaging, respectively. This arrangement is designed to ensure that only light transmitted from the Multi-Mode Illumination Subsystem **14** and reflected from the illuminated object is ultimately transmitted through a narrow-band transmission optical filter subsystem **4** realized by (1) high-pass (i.e. red-wavelength reflecting) filter element **4**A mounted at the light transmission aperture **3** immediately in front of panel **5**, and (2) low-pass filter element **4**B mounted either before the image sensing array **22** or anywhere after panel **5** as shown in FIG. **3**C. FIG. **6**A**4** sets forth the resulting composite transmission characteristics of the narrow-band transmission spectral filter subsystem **4**, plotted against the spectral characteristics of the emission from the LED illumination arrays employed in the Multi-Mode Illumination Subsystem **14**.

The primary function of the narrow-band integrated optical filter subsystem **4** is to ensure that the CMOS image sensing array **22** only receives the narrow-band visible illumination transmitted by the three sets of LED-based illumination arrays **27**, **28** and **29** driven by LED driver circuitry **30** associated with the Multi-Mode Illumination Subsystem **14**, whereas all other components of ambient light collected by

the light collection optics are substantially rejected at the image sensing array **22**, thereby providing improved SNR thereat, thus improving the performance of the system.

The primary function of the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** is to twofold: (1) to measure, in real-time, the power density [joules/cm] of photonic energy (i.e. light) collected by the optics of the system at about its image sensing array **22**, and generate Auto-Exposure Control Signals indicating the amount of exposure required for good image formation and detection; and (2) in combination with Illumination Array Selection Control Signal provided by the System Control Subsystem **19**, automatically drive and control the output power of selected LED arrays **27**, **28** and/or **29** in the Multi-Mode Illumination Subsystem, so that objects within the FOV of the system are optimally exposed to LED-based illumination and optimal images are formed and detected at the image sensing array **22**.

The primary function of the Image Capturing and Buffering Subsystem **16** is to (1) detect the entire 2-D image focused onto the 2D image sensing array **22** by the image formation optics **21** of the system, (2) generate a frame of digital pixel data **31** for either a selected region of interest of the captured image frame, or for the entire detected image, and then (3) buffer each frame of image data as it is captured. Notably, in the illustrative embodiment, a single 2D image frame (**31**) is captured during each image capture and processing cycle, or during a particular stage of a processing cycle, so as to eliminate the problems associated with image frame overwriting, and synchronization of image capture and decoding processes, as addressed in U.S. Pat. Nos. 5,932,862 and 5,942,741 assigned to Welch Allyn, and incorporated herein by reference.

The primary function of the Multi-Mode Imaging-Based Bar Code Symbol Reading Subsystem **17** is to process images that have been captured and buffered by the Image Capturing and Buffering Subsystem **16**, during both narrow-area and wide-area illumination modes of system operation. Such image processing operation includes image-based bar code decoding methods illustrated in FIGS. **14** through **25**, and described in detail hereinafter.

The primary function of the Input/Output Subsystem **18** is to support standard and/or proprietary communication interfaces with external host systems and devices, and output processed image data and the like to such external host systems or devices by way of such interfaces. Examples of such interfaces, and technology for implementing the same, are given in U.S. Pat. No. 6,619,549, incorporated herein by reference in its entirety.

The primary function of the System Control Subsystem **19** is to provide some predetermined degree of control or management signaling services to each subsystem component integrated, as shown. While this subsystem can be implemented by a programmed microprocessor, in the illustrative embodiment, it is implemented by the three-tier software architecture supported on computing platform shown in FIG. **2**B, and as represented in FIGS. **11**A through **13**L, and described in detail hereinafter.

The primary function of the manually-activatable Trigger Switch **2**C integrated with the hand-supportable housing is to enable the user to generate a control activation signal upon manually depressing the Trigger Switch **2**C, and to provide this control activation signal to the System Control Subsystem **19** for use in carrying out its complex system and subsystem control operations, described in detail herein.

The primary function of the System Mode Configuration Parameter Table **70** is to store (in non-volatile/persistent memory) a set of configuration parameters for each of the available Programmable Modes of System Operation specified in the Programmable Mode of Operation Table shown in FIGS. **26**A through **26**B and which can be read and used by the System Control Subsystem **19** as required during its complex operations.

The detailed structure and function of each subsystem will now be described in detail above.

Schematic Diagram as System Implementation Model for the Hand-Supportable Digital Imaging-Based Bar Code Reading Device of the Present Invention

FIG. **2**B shows a schematic diagram of a system implementation for the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device **1** illustrated in FIGS. **1**A through **1**L. As shown in this system implementation, the bar code symbol reading device is realized using a number of hardware component comprising: an illumination board **33** carrying components realizing electronic functions performed by the LED-Based Multi-Mode Illumination Subsystem **14** and Automatic Light Exposure Measurement And Illumination Control Subsystem **15**; a CMOS camera board **34** carrying high resolution (1280×1024 8-bit 6 micron pixel size) CMOS image sensing array **22** running at 25 Mhz master clock, at 7 frames/second at 1280*1024 resolution with randomly accessible region of interest (ROI) window capabilities, realizing electronic functions performed by the Multi-Mode Image Formation and Detection Subsystem **13**; a CPU board **35** (i.e. computing platform) including (i) an Intel Sabinal 32-Bit Microprocessor PXA210 **36** running at 200 mHz 1.0 core voltage with a 16 bit 100 Mhz external bus speed, (ii) an expandable (e.g. 8+ megabyte) Intel J3 Asynchronous 16-bit Flash memory **37**, (iii) an 16 Megabytes of 100 MHz SDRAM **38**, (iv) an Xilinx Spartan II FPGA FIFO **39** running at 50 Mhz clock frequency and 60 MB/Sec data rate, configured to control the camera timings and drive an image acquisition process, (v) a multimedia card socket **40**, for realizing the other subsystems of the system, (vi) a power management module **41** for the MCU adjustable by the I2C bus, and (vii) a pair of UARTs **42**A and **42**B (one for an IRDA port and one for a JTAG port); an interface board **43** for realizing the functions performed by the I/O subsystem **18**; and an IR-based object presence and range detection circuit **44** for realizing Subsystem **12**.

In the illustrative embodiment, the image formation optics **21** supported by the bar code reader provides a field of view of 103 mm at the nominal focal distance to the target, of approximately 70 mm from the edge of the bar code reader. The minimal size of the field of view (FOV) is 62 mm at the nominal focal distance to the target of approximately 10 mm. Preliminary tests of the parameters of the optics are shown on FIG. **4**B (the distance on FIG. **4**B is given from the position of the image sensing array **22**, which is located inside the bar code symbol reader approximately 80 mm from the edge). As indicated in FIG. **4**C, the depth of field of the image formation optics varies from approximately 69 mm for the bar codes with resolution of 5 mils per narrow module, to 181 mm for the bar codes with resolution of 13 mils per narrow module.

The Multi-Mode Illumination Subsystem **14** is designed to cover the optical field of view (FOV) **23** of the bar code symbol reader with sufficient illumination to generate high-contrast images of bar codes located at both short and long distances from the imaging window. The illumination subsystem also provides a narrow-area (thin height) targeting beam **24** having dual purposes: (a) to indicate to the user where the optical view of the reader is; and (b) to allow a quick scan of just a few lines of the image and attempt a super-fast

bar code decoding if the bar code is aligned properly. If the bar code is not aligned for a linearly illuminated image to decode, then the entire field of view is illuminated with a wide-area illumination field **25** or **26** and the image of the entire field of view is acquired by Image Capture and Buffering Subsystem **16** and processed by Multi-Mode Bar Code Symbol Reading Subsystem **17**, to ensure reading of a bar code symbol presented therein regardless of its orientation.

The interface board **43** employed within the bar code symbol reader provides the hardware communication interfaces for the bar code symbol reader to communicate with the outside world. The interfaces implemented in system will typically include RS232, keyboard wedge, and/or USB, or some combination of the above, as well as others required or demanded by the particular application at hand.

Specification of the Area-Type Image Formation and Detection (i.e. Camera) Subsystem During its Narrow-Area (Linear) and Wide-Area Modes of Imaging, Supported by the Narrow and Wide Area Fields of Narrow-Band Illumination, Respectively

As shown in FIGS. **3**B through **3**E, the Multi-Mode Image Formation And Detection (IFD) Subsystem **13** has a narrow-area image capture mode (i.e. where only a few central rows of pixels about the center of the image sensing array are enabled) and a wide-area image capture mode of operation (i.e. where all pixels in the image sensing array are enabled). The CMOS image sensing array **22** in the Image Formation and Detection Subsystem **13** has image formation optics **21** which provides the image sensing array with a field of view (FOV) **23** on objects to be illuminated and imaged. As shown, this FOV is illuminated by the Multi-Mode Illumination Subsystem **14** integrated within the bar code reader.

The Multi-Mode Illumination Subsystem **14** includes three different LED-based illumination arrays **27**, **28** and **29** mounted on the light transmission window panel **5**, and arranged about the light transmission window **4**A. Each illumination array is designed to illuminate a different portion of the FOV of the bar code reader during different modes of operation. During the narrow-area (linear) illumination mode of the Multi-Mode Illumination Subsystem **14**, the central narrow-wide portion of the FOV indicated by **23** is illuminated by the narrow-area illumination array **27**, shown in FIG. **3**A. During the near-field wide-area illumination mode of the Multi-Mode Illumination Subsystem **14**, which is activated in response to the IR Object Presence and Range Detection Subsystem **12** detecting an object within the near-field portion of the FOV, the near-field wide-area portion of the FOV is illuminated by the near-field wide-area illumination array **28**, shown in FIG. **3**A. During the far-field wide-area illumination mode of the Multi-Mode Illumination Subsystem **14**, which is activated in response to the IR Object Presence and Range Detection Subsystem **12** detecting an object within the far-field portion of the FOV, the far-field wide-area portion of the FOV is illuminated by the far-field wide-area illumination array **29**, shown in FIG. **3**A. In FIG. **3**A, the spatial relationships are shown between these fields of narrow-band illumination and the far and near field portions the FOV of the Image Formation and Detection Subsystem **13**.

In FIG. **3**B, the Multi-Mode LED-Based Illumination Subsystem **14** is shown transmitting visible narrow-band illumination through its narrow-band transmission-type optical filter subsystem **4**, shown in FIG. **3**C and integrated within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device. The narrow-band illumination from the Multi-Mode Illumination Subsystem **14** illuminates an object with the FOV of the image formation optics of the Image Formation and Detection Subsystem **13**, and light rays reflected and scattered therefrom are transmitted through the high-pass and low-pass optical filters **4**A and **4**B and are ultimately focused onto image sensing array **22** to form of a focused detected image thereupon, while all other components of ambient light are substantially rejected before reaching image detection at the image sensing array **22**. Notably, in the illustrative embodiment, the red-wavelength reflecting high-pass optical filter element **4**A is positioned at the imaging window of the device before the image formation optics **21**, whereas the low-pass optical filter element **4**B is disposed before the image sensing array **22** between the focusing lens elements of the image formation optics **21**. This forms narrow-band optical filter subsystem **4** which is integrated within the bar code reader to ensure that the object within the FOV is imaged at the image sensing array **22** using only spectral components within the narrow-band of illumination produced from Subsystem **14**, while rejecting substantially all other components of ambient light outside this narrow range (e.g. 15 nm).

As shown in FIG. **3**D, the Image Formation And Detection Subsystem **14** employed within the hand-supportable image-based bar code reading device comprising three lenses **21**A, **21**B and **21**C, each made as small as possible (with a maximum diameter of 12 mm), having spherical surfaces, and made from common glass, e.g. LAK2 (~LaK9), ZF10 (=SF8), LAF2 (~LaF3). Collectively, these lenses are held together within a lens holding assembly **45**, as shown in FIG. **3**E, and form an image formation subsystem arranged along the optical axis of the CMOS image sensing array **22** of the bar code reader.

As shown in FIG. **3**E, the lens holding assembly **45** comprises: a barrel structure **45A1**, **45A2** for holding lens elements **21**A, **21**B and **21**C; and a base structure **45**B for holding the image sensing array **22**; wherein the assembly is configured so that the barrel structure **45**A slides within the base structure **45**B so as to focus the fixed-focus lens assembly during manufacture.

In FIGS. **3**F**1** and **3**F**2**, the lens holding assembly **45** and imaging sensing array **22** are mounted along an optical path defined along the central axis of the system. In the illustrative embodiment, the image sensing array **22** has, for example, a 1280×1024 pixel resolution (½" format), 6 micron pixel size, with randomly accessible region of interest (ROI) window capabilities. It is understood, though, that many others kinds of imaging sensing devices (e.g. CCD) can be used to practice the principles of the present invention disclosed herein, without departing from the scope or spirit of the present invention.

Method of Designing the Image Formation (i.e. Camera) Optics within the Image-Based Bar Code Reader of the Present Invention Using The Modulation Transfer Function (MTF)

The function of the image formation (i.e. camera) optics in the Image Formation and Detection Subsystem **13** is to form and project, as accurately as possible, an image of the object being formed on the image sensing array **22**. In practice, it is impossible to get an absolutely perfect image reproduction of the object with no loss of information, because the quality of the image is limited by various effects. These effects include: (i) diffraction, always present in even the very best lenses; (ii) aberrations which, if present, can generally only be minimized, not eliminated; (iii) variation of the distance to the object, especially if the lens cannot dynamically adjust its focus; and so on. Before spending time and money to produce a lens assembly, it is necessary to determine that a given lens design for the bar code symbol reader of the present invention

will perform well enough to satisfy the requirements of the application. Thus, it will be extremely helpful to (i) establish one or more design criteria to quantify the lens performance, and (ii) optimize the design around these criteria until the desired performance is achieved.

The preferred criterion for designing the image formation optics in the system hereof is the modulation transfer function, or MTF. The MTF provides a measure of the contrast present in an object or image. Qualitatively, contrast may be thought of as the difference between light and dark regions in the object or image. The greater the difference in "brightness" between two regions of the object or image, the greater the contrast.

Considering the image, given the data from the image sensor, a quantitative treatment is possible.

On the common 8 bit scale, a pixel that is totally black is assigned the value 0, while a pixel that is totally saturated white is assigned the value **255**.

Also, the closer the spacing of the object features, then the worse the reproduction of that contrast in the image of the object.

A mathematical expression is required to quantify the amount of contrast present in an object or image, so that its variation after imaging through the optics may be assessed. A useful contrast measure can be defined as the modulation M of a given region in the object, given as follows:

$$M = \frac{\text{max value} - \text{min value}}{\text{max value} + \text{min value}}$$

The greater the contrast in the object or image, the greater the value of M, up to a maximum of 1. On the other hand, no contrast whatever in the object or image (i.e. no distinguishable features in the region of the object in question) yields a modulation of 0. To determine how well the image formation optics preserves the modulation of the target object in the image, it is only necessary to form a ratio of the image modulation to the object modulation, which is the MTF:

$$MTF = \frac{\text{image modulation}}{\text{object modulation}}$$

Perfect reproduction of the object contrast in the image (impossible in practice) results in an MTF of 1. A total loss of the object contrast in the image gives an MTF of 0.

The MTF is a useful concept in optical design because it simultaneously accounts for the impact of any effect that degrades the quality of the image, usually referred to as blurring. As described previously, these effects include diffraction, aberrations (spherical, chromatic, coma, astigmatism, field curvature) and deviation of the object distance from its nominal value. It should be mentioned for sake of completeness, however, that MTF is not a single perfect or all-encompassing measure of image quality. One potential drawback is that examining the MTF reveals only the total impact of all effects simultaneously, and cannot distinguish between blurring caused by one defect or another. If it is necessary to determine what effects are degrading the MTF, and to what extent for each, then other methods must be used, and other criteria examined. In addition, there are potentially negative image characteristics, such as distortion, that are not revealed at all by the MTF. If the optical designer is not careful, then it is possible that an image with an MTF close to the diffraction limit, which is as good as it is possible to get, may have distortion so bad that it is unusable in the application at hand.

In accordance with the design method of the present invention, after calculating the MTF for a given optical design, an additional criterion is necessary to specify what MTF is good enough for the application in question. For bar code decoding applications, a useful rule of thumb is that 0.3 MTF or better is needed for decoding software to work reliably well in an Imaging-Based Bar Code Symbol Reader. The design strategy employed on the Imaging-Based Bar Code Symbol Reader of the present invention is to determine, as a function of object distance, the code element size (in millimeters) at which the MTF of the resulting image falls to 0.3. In other words, at each object distance, the optical designer should determine what is the smallest size of code element (in millimeters) that can be imaged well enough to be read by the Multi-Mode Image-Processing Bar Code Reading Subsystem **17** of the present invention. At one stage of the design of the image formation optics employed in the illustrative embodiment, the plot of minimum code element size against object distance is generated, as shown in FIG. **4**E.

Given such a plot, the optical design team needs to determine whether or not the resulting bar code reader performance satisfies the requirements of the application at hand. To help make this determination, an advanced optical design method and tool described below can be used with excellent results.

Method of Theoretically Characterizing the DOF of the Image Formation Optics Employed in the Imaging-Based Bar Code Reader of the Present Invention

Referring to FIGS. **4**D through **4I3**, a novel software-enabled design tool and method will now be described.

In general, the software-enabled optical design tool provides a novel way of and means for completely theoretically characterizing, and graphically viewing and interpreting the composite DOF of image formation optics (e.g. such as **21** employed in the Imaging-Based Bar Code Symbol Reader of the present invention) as well as other imaging-based optical readers, while simultaneously accounting for optical performance and image sensor limitations, over all desired object distances and for all desired code mil sizes.

Given an arrangement of lens elements for the design of the image formation optics **21**, the optical design method of the present invention involves using a software-based optical design tool, as described in FIGS. **4I1** through **4I3**, to generate the composite DOF chart in accordance with the present invention. The functions required by this optical design tool will be described below. The software-based optical design tool (i.e. computer program) of the illustrative embodiment, described in FIGS. **4I1** through **4I3**, has been developed using Zemax® optical modeling software, programmed in ZPL (Zemax Programming Language) in accordance with the principles of the present invention described in detail below.

The first function required by the optical design tool of the present invention is that it must be able to calculate the modulation transfer function (MTF) of the image resulting from image formation optics **21**, plotted as a function of object distance. The general industry rule of thumb is that a 0.3 MTF is the minimum acceptable for bar code decoding. Therefore, this software-based optical design tool must be able to determine, as a function of object distance, the object spatial-frequency at which the MTF of the image drops to 0.3.

The second function required by the optical design tool of the present invention is that it must be able to convert the object spatial-frequency to code mil size, and then this data should be plotted against object distance. A resulting plot is shown in FIG. **4**D, where the dotted-line curve shows the

optical performance of the image formation optics, in terms of the smallest mil size code that can be decoded, at a given object distance. FIG. **4**E demonstrates how to read the DOF from this plot, by finding the intersections of the mil size in question with the optical performance curve.

However, optical performance of the image formation optics is not the only factor determining the capacity of an Imaging-Based Bar Code Symbol Reader to read bar code symbols having bar code elements of a given width. Image-processing based bar code symbol decoding software requires a certain minimum number of sensor pixel "fields of view" to be projected onto each minimum width code element, within the field of the view of the image formation optics. The general industry rule of thumb is that 1.6 pixels are required per narrow element for acceptable decoding. In accordance with the present invention, this rule has been expanded to the range of 1.4 to 1.6 pixels per narrow element, and can be considered a limit imposed by sampling theory that will restrict the ultimate performance of the bar code symbol reader **1** regardless of the individual performance of its image formation optics **21**.

Therefore, the third function required by the optical design tool of the present invention is that it must be able to calculate, as a function of object distance, the size of the field of view of a single sensor pixel when projected through the image formation optics **21** and out into object space (that is, accounting for the optical magnification of the image formation optics **21**). These linear functions, both for the 1.4 and 1.6 pixel rules, are preferably plotted on the same axes as the optical performance curve, as shown in FIG. **4**F.

Having described the primary functionalities of the optical design tool of the present invention, and how to generate a composite DOF plot as shown in FIG. **4**F for an Imaging-Based Bar Code Symbol Reader, it is now appropriate to describe, with reference to FIG. **4**G, how to determine the actual composite DOF curve, accounting for both optical performance and sampling limit, for the 1.6 pixel case. Other system information, such as lens focal length, lens f-number, etc. may also be displayed on the composite DOF plot of FIG. **4**G, for instance in a title block.

As shown in FIG. **4**G, the method involves following the optical performance curve until it intersects the sampling limit line. Then, the sampling limit line is followed until it re-intersects the optical performance curve, at which point the optical performance curve is again followed. Thus, the sampling limit line of choice represents the lower limit of the decoding resolution of the system. Referring to FIG. **4**H, a simple technique is shown for reading the DOF from the composite plot of FIG. **4**G.

Preferably, the optical design tool of the present invention will be provide with a simple graphical user interface (GUI) may be useful, supporting pop-up windows to enable the user to easily type numbers into the program. Also, the optical design tool will preferably implement various methods to allow the user to specify some of the required numbers while the program is running, as oppose to having to change the numbers in the program file.

A less preferred alternative way of practicing the optical design method of the present invention would be to manually construct the composite DOF plot by examining MTF data and plotting the results in Excel, for example. However, this approach is labor intensive and does not offer any appreciable increase in accuracy, as does the use of the software-enabled optical design tool described in FIGS. **4I1** through **4I3**.

Specification of Multi-Mode LED-Based Illumination Subsystem Employed in the Hand-Supportable Image-Based Bar Code Reading System of the Present Invention

In the illustrative embodiment, the LED-Based Multi-Mode Illumination Subsystem **14** comprises: narrow-area illumination array **27**; near-field wide-area illumination array **28**; and far-field wide-area illumination array **29**. The three fields of narrow-band illumination produced by the three illumination arrays of subsystem **14** are schematically depicted in FIG. **5A1**. As will be described hereinafter, with reference to FIGS. **27** and **28**, narrow-area illumination array **27** can be realized as two independently operable arrays, namely: a near-field narrow-area illumination array and a far-field narrow-area illumination array, which are activated when the target object is detected within the near and far fields, respectively, of the automatic IR-based Object Presence and Range Detection Subsystem **12** during wide-area imaging modes of operation. However, for purposes of illustration, the first illustrative embodiment of the present invention employs only a single field narrow-area (linear) illumination array which is designed to illuminate over substantially entire working range of the system, as shown in FIG. **5A1**.

As shown in FIGS. **5**B, **5C3** and **5C4**, the narrow-area (linear) illumination array **27** includes two pairs of LED light sources **27A1** and **27A2** provided with cylindrical lenses **27B1** and **27B2**, respectively, and mounted on left and right portions of the light transmission window panel **5**. During the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, the narrow-area (linear) illumination array **27** produces narrow-area illumination field **24** of narrow optical-bandwidth within the FOV of the system. In the illustrative embodiment, narrow-area illumination field **24** has a height less than 10 mm at far field, creating the appearance of substantially linear or rather planar illumination field.

The near-field wide-area illumination array **28** includes two sets of (flattop) LED light sources **28A1-28A6** and **28A7-28A13** without any lenses mounted on the top and bottom portions of the light transmission window panel **5**, as shown in FIG. **5**B. During the near-field wide-area image capture mode of the Image Formation and Detection Subsystem **13**, the near-field wide-area illumination array **28** produces a near-field wide-area illumination field **25** of narrow optical-bandwidth within the FOV of the system.

As shown in FIGS. **5**B, **5D3** and **5D4**, the far-field wide-area illumination array **29** includes two sets of LED light sources **29A1-29A6** and **29A7-29A13** provided with spherical (i.e. plano-convex) lenses **29B1-29B6** and **29B7-29B13**, respectively, and mounted on the top and bottom portions of the light transmission window panel **5**. During the far-field wide-area image capture mode of the Image Formation and Detection Subsystem **13**, the far-field wide-area illumination array **29** produces a far-field wide-area illumination beam of narrow optical-bandwidth within the FOV of the system.

Narrow-Area (Linear) Illumination Arrays Employed in the Multi-Mode Illumination Subsystem

As shown in FIG. **5A1**, the narrow-area (linear) illumination field **24** extends from about 30 mm to about 200 mm within the working range of the system, and covers both the near and far fields of the system. The near-field wide-area illumination field **25** extends from about 0 mm to about 100 mm within the working range of the system. The far-field wide-area illumination field **26** extends from about 100 mm to about 200 mm within the working range of the system. The Table shown in FIG. **5A2** specifies the geometrical properties

and characteristics of each illumination mode supported by the Multi-Mode LED-based Illumination Subsystem **14** of the present invention.

The narrow-area illumination array **27** employed in the Multi-Mode LED-Based Illumination Subsystem **14** is optically designed to illuminate a thin area at the center of the field of view (FOV) of the Imaging-Based Bar Code Symbol Reader, measured from the boundary of the left side of the field of view to the boundary of its right side, as specified in FIG. **5A1**. As will be described in greater detail hereinafter, the narrow-area illumination field **24** is automatically generated by the Multi-Mode LED-Based Illumination Subsystem **14** in response to the detection of an object within the object detection field of the automatic IR-based Object Presence and Range Detection Subsystem **12**. In general, the object detection field of the IR-based Object Presence and Range Detection Subsystem **12** and the FOV of the Image Formation and Detection Subsystem **13** are spatially co-extensive and the object detection field spatially overlaps the FOV along the entire working distance of the Imaging-Based Bar Code Symbol Reader. The narrow-area illumination field **24**, produced in response to the detection of an object, serves a dual purpose: it provides a visual indication to an operator about the location of the optical field of view of the bar code symbol reader, thus, serves as a field of view aiming instrument; and during its image acquisition mode, the narrow-area illumination beam is used to illuminated a thin area of the FOV within which an object resides, and a narrow 2-D image of the object can be rapidly captured (by a small number of rows of pixels in the image sensing array **22**), buffered and processed in order to read any linear bar code symbols that may be represented there within.

FIG. **5C1** shows the Lambertian emittance versus wavelength characteristics of the LEDs used to implement the narrow-area illumination array **27** in the Multi-Mode Illumination Subsystem **14**. FIG. **5C2** shows the Lambertian emittance versus polar angle characteristics of the same LEDs. FIG. **5C3** shows the cylindrical lenses used before the LEDs (633 nm InGaAlP) in the narrow-area (linear) illumination arrays in the illumination subsystem of the present invention. As shown, the first surface of the cylindrical lens is curved vertically to create a narrow-area (linear) illumination pattern, and the second surface of the cylindrical lens is curved horizontally to control the height of the of the linear illumination pattern to produce a narrow-area illumination pattern. FIG. **5C4** shows the layout of the pairs of LEDs and two cylindrical lenses used to implement the narrow-area illumination array of the illumination subsystem of the present invention. In the illustrative embodiment, each LED produces about a total output power of about 11.7 mW under typical conditions. FIG. **5C5** sets forth a set of six illumination profiles for the narrow-area illumination fields produced by the narrow-area illumination arrays of the illustrative embodiment, taken at 30, 40, 50, 80, 120, and 220 millimeters along the field away from the imaging window (i.e. working distance) of the bar code reader of the present invention, illustrating that the spatial intensity of the area-area illumination field begins to become substantially uniform at about 80 millimeters. As shown, the narrow-area illumination beam is usable beginning 40 mm from the light transmission/imaging window.

Near-Field Wide-Area Illumination Arrays Employed in the Multi-Mode Illumination Subsystem

The near-field wide-area illumination array **28** employed in the LED-Based Multi-Mode Illumination Subsystem **14** is optically designed to illuminate a wide area over a near-field portion of the field of view (FOV) of the Imaging-Based Bar Code Symbol Reader, as defined in FIG. **5A1**. As will be described in greater detail hereinafter, the near-field wide-area illumination field **28** is automatically generated by the LED-based Multi-Mode Illumination Subsystem **14** in response to: (1) the detection of any object within the near-field of the system by the IR-based Object Presence and Range Detection Subsystem **12**; and (2) one or more of following events, including, for example: (i) failure of the image processor to successfully decode process a linear bar code symbol during the narrow-area illumination mode; (ii) detection of code elements such as control words associated with a 2-D bar code symbol; and/or (iii) detection of pixel data in the image which indicates that object was captured in a state of focus.

In general, the object detection field of the IR-based Object Presence and Range Detection Subsystem **12** and the FOV of the Image Formation And Detection Subsystem **13** are spatially co-extensive and the object detection field spatially overlaps the FOV along the entire working distance of the Imaging-Based Bar Code Symbol Reader. The near-field wide-area illumination field **23**, produced in response to one or more of the events described above, illuminates a wide area over a near-field portion of the field of view (FOV) of the Imaging-Based Bar Code Symbol Reader, as defined in FIG. **5A**, within which an object resides, and a 2-D image of the object can be rapidly captured (by all rows of the image sensing array **22**, buffered and decode-processed in order to read any 1D or 2-D bar code symbols that may be represented there within, at any orientation, and of virtually any bar code symbology. The intensity of the near-field wide-area illumination field during object illumination and image capture operations is determined by how the LEDs associated with the near-field wide array illumination arrays **28** are electrically driven by the Multi-Mode Illumination Subsystem **14**. The degree to which the LEDs are driven is determined by the intensity of reflected light measured near the image formation plane by the automatic light exposure and control subsystem **15**. If the intensity of reflected light at the photodetector of the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** is weak, indicative that the object exhibits low light reflectivity characteristics and a more intense amount of illumination will need to be produced by the LEDs to ensure sufficient light exposure on the image sensing array **22**, then the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** will drive the LEDs more intensely (i.e. at higher operating currents).

FIG. **5D1** shows the Lambertian emittance versus wavelength characteristics of the LEDs used to implement the wide area illumination arrays in the illumination subsystem of the present invention. FIG. **5D2** shows the Lambertian emittance versus polar angle characteristics of the LEDs used to implement the near field wide-area illumination arrays in the Multi-Mode Illumination Subsystem **14**. FIG. **5D4** is geometrical the layout of LEDs used to implement the narrow wide-area illumination array of the Multi-Mode Illumination Subsystem **14**, wherein the illumination beam produced therefrom is aimed by angling the lenses before the LEDs in the near-field wide-area illumination arrays of the Multi-Mode Illumination Subsystem **14**. FIG. **5D5** sets forth a set of six illumination profiles for the near-field wide-area illumination fields produced by the near-field wide-area illumination arrays of the illustrative embodiment, taken at 10, 20, 30, 40, 60, and 100 millimeters along the field away from the imaging window (i.e. working distance) of the Imaging-Based Bar Code Symbol Reader **1**. These plots illustrate that the spatial intensity of the near-field wide-area illumination

field begins to become substantially uniform at about 40 millimeters (i.e. center:edge=2:1 max).

Far-Field Wide-Area Illumination Arrays Employed in the Multi-Mode Illumination Subsystem

The far-field wide-area illumination array **26** employed in the Multi-Mode LED-based Illumination Subsystem **14** is optically designed to illuminate a wide area over a far-field portion of the field of view (FOV) of the Imaging-Based Bar Code Symbol Reader, as defined in FIG. **5A1**. As will be described in greater detail hereinafter, the far-field wide-area illumination field **26** is automatically generated by the LED-Based Multi-Mode Illumination Subsystem **14** in response to: (1) the detection of any object within the near-field of the system by the IR-based Object Presence and Range Detection Subsystem **12**; and (2) one or more of following events, including, for example: (i) failure of the image processor to successfully decode process a linear bar code symbol during the narrow-area illumination mode; (ii) detection of code elements such as control words associated with a 2-D bar code symbol; and/or (iii) detection of pixel data in the image which indicates that object was captured in a state of focus. In general, the object detection field of the IR-based Object Presence and Range Detection Subsystem **12** and the FOV **23** of the image detection and formation subsystem **13** are spatially co-extensive and the object detection field **20** spatially overlaps the FOV **23** along the entire working distance of the Imaging-Based Bar Code Symbol Reader. The far-field wide-area illumination field **26**, produced in response to one or more of the events described above, illuminates a wide area over a far-field portion of the field of view (FOV) of the Imaging-Based Bar Code Symbol Reader, as defined in FIG. **5**A, within which an object resides, and a 2-D image of the object can be rapidly captured (by all rows of the image sensing array **22**), buffered and processed in order to read any 1D or 2-D bar code symbols that may be represented there within, at any orientation, and of virtually any bar code symbology. The intensity of the far-field wide-area illumination field during object illumination and image capture operations is determined by how the LEDs associated with the far-field wide-area illumination array **29** are electrically driven by the Multi-Mode Illumination Subsystem **14**. The degree to which the LEDs are driven (i.e. measured in terms of junction current) is determined by the intensity of reflected light measured near the image formation plane by the Automatic Light Exposure Measurement And Illumination Control Subsystem **15**. If the intensity of reflected light at the photo-detector of the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** is weak, indicative that the object exhibits low light reflectivity characteristics and a more intense amount of illumination will need to be produced b the LEDs to ensure sufficient light exposure on the image sensing array **22**, then the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** will drive the LEDs more intensely (i.e. at higher operating currents).

During both near and far field wide-area illumination modes of operation, the Automatic Light Exposure Measurement and Illumination Control Subsystem (i.e. module) **15** measures and controls the time duration which the Multi-Mode Illumination Subsystem **14** exposes the image sensing array **22** to narrow-band illumination (e.g. 633 nanometers, with approximately 15 nm bandwidth) during the image capturing/acquisition process, and automatically terminates the generation of such illumination when such computed time duration expires. In accordance with the principles of the present invention, this global exposure control process ensures that each and every acquired image has good contrast and is not saturated, two conditions essential for consistent and reliable bar code reading

FIG. **5D1** shows the Lambertian emittance versus wavelength characteristics of the LEDs used to implement the far-field wide-area illumination arrays **29** in the Multi-Mode Illumination Subsystem **14** FIG. **5D2** shows the Lambertian emittance versus polar angle characteristics of the LEDs used to implement the same. FIG. **5D3** shows the piano-convex lenses used before the LEDs in the far-field wide-area illumination arrays in the Multi-Mode Illumination Subsystem **14**. FIG. **5D4** shows a layout of LEDs and plano-convex lenses used to implement the far wide-area illumination array **29** of the illumination subsystem, wherein the illumination beam produced therefrom is aimed by angling the lenses before the LEDs in the far-field wide-area illumination arrays of the Multi-Mode Illumination Subsystem **14**. FIG. **5D6** sets forth a set of three illumination profiles for the far-field wide-area illumination fields produced by the far-field wide-area illumination arrays of the illustrative embodiment, taken at 100, 150 and 220 millimeters along the field away from the imaging window (i.e. working distance) of the Imaging-Based Bar Code Symbol Reader **1**, illustrating that the spatial intensity of the far-field wide-area illumination field begins to become substantially uniform at about 100 millimeters. FIG. **5D7** shows a table illustrating a preferred method of calculating the pixel intensity value for the center of the far field wide-area illumination field produced from the Multi-Mode Illumination Subsystem **14**, showing a significant signal strength (greater than 80DN at the far center field).

Specification of the Narrow-Band Optical Filter Subsystem Integrated within the Hand-Supportable Housing of the Imager of the Present Invention

As shown in FIG. **6A1**, the hand-supportable housing of the bar code reader of the present invention has integrated within its housing, narrow-band optical filter subsystem **4** for transmitting substantially only the very narrow band of wavelengths (e.g. 620-700 nanometers) of visible illumination produced from the narrow-band Multi-Mode Illumination Subsystem **14**, and rejecting all other optical wavelengths outside this narrow optical band however generated (i.e. ambient light sources). As shown, narrow-band optical filter subsystem **4** comprises: red-wavelength reflecting (high-pass) imaging window filter **4**A integrated within its light transmission aperture **3** formed on the front face of the hand-supportable housing; and low pass optical filter **4**B disposed before the CMOS image sensing array **22**. These optical filters **4**A and **4**B cooperate to form the narrow-band optical filter subsystem **4** for the purpose described above. As shown in FIG. **6A2**, the light transmission characteristics (energy versus wavelength) associated with the low-pass optical filter element **4**B indicate that optical wavelengths below 620 nanometers are transmitted there through, whereas optical wavelengths above 620 nm are substantially blocked (e.g. absorbed or reflected). As shown in FIG. **6A3**, the light transmission characteristics (energy versus wavelength) associated with the high-pass imaging window filter **4**A indicate that optical wavelengths above 700 nanometers are transmitted there through, thereby producing a red-color appearance to the user, whereas optical wavelengths below 700 nm are substantially blocked (e.g. absorbed or reflected) by optical filter **4**A.

During system operation, spectral band-pass filter subsystem **4** greatly reduces the influence of the ambient light, which falls upon the CMOS image sensing array **22** during the image capturing operations. By virtue of the optical filter

of the present invention, a optical shutter mechanism is eliminated in the system. In practice, the optical filter can reject more than 85% of incident ambient light, and in typical environments, the intensity of LED illumination is significantly more than the ambient light on the CMOS image sensing array **22**. Thus, while an optical shutter is required in nearly most conventional CMOS imaging systems, the imaging-based bar code reading system of the present invention effectively manages the exposure time of narrow-band illumination onto its CMOS image sensing array **22** by simply controlling the illumination time of its LED-based illumination arrays **27**, **28** and **29** using control signals generated by Automatic Light Exposure Measurement and Illumination Control Subsystem **15** and the CMOS image sensing array **22** while controlling illumination thereto by way of the band-pass optical filter subsystem **4** described above. The result is a simple system design, without moving parts, and having a reduced manufacturing cost.

While the band-pass optical filter subsystem **4** is shown comprising a high-pass filter element **4**A and low-pass filter element **4**B, separated spatially from each other by other optical components along the optical path of the system, subsystem **4** may be realized as an integrated multi-layer filter structure installed in front of the Image Formation And Detection (IFD) Module **13**, or before its image sensing array **22**, without the use of the high-pass window filter **4**A, or with the use thereof so as to obscure viewing within the Imaging-Based Bar Code Symbol Reader while creating an attractive red-colored protective window. Preferably, the red-color window filter **4**A will have substantially planar surface characteristics to avoid focusing or defocusing of light transmitted there through during imaging operations.

Specification of the Automatic Light Exposure Measurement and Illumination Control Subsystem of the Present Invention

The primary function of the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** is to control the brightness and contrast of acquired images by (i) measuring light exposure at the image plane of the CMOS imaging sensing array **22** and (ii) controlling the time duration that the Multi-Mode Illumination Subsystem **14** illuminates the target object with narrow-band illumination generated from the activated LED illumination array. Thus, the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** eliminates the need for a complex shuttering mechanism for CMOS-based image sensing array **22**. This novel mechanism ensures that the Imaging-Based Bar Code Symbol Reader of the present invention generates non-saturated images with enough brightness and contrast to guarantee fast and reliable image-based bar code decoding in demanding end-user applications.

During object illumination, narrow-band LED-based light is reflected from the target object (at which the hand-supportable bar code reader is aimed) and is accumulated by the CMOS image sensing array **22**. Notably, the object illumination process must be carried out for an optimal duration so that the acquired image frame has good contrast and is not saturated. Such conditions are required for the consistent and reliable bar code decoding operation and performance. The Automatic Light Exposure Measurement and Illumination Control Subsystem **15** measures the amount of light reflected from the target object, calculates the maximum time that the CMOS image sensing array **22** should be kept exposed to the actively-driven LED-based illumination array associated with the Multi-Mode Illumination Subsystem **14**, and then automatically deactivates the illumination array when the calculated time to do so expires (i.e. lapses).

As shown in FIG. **7**A of the illustrative embodiment, the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** comprises: a parabolic light-collecting mirror **55** mounted within the head portion of the hand-supportable housing, for collecting narrow-band LED-based light reflected from a central portion of the FOV of the system, which is then transmitted through the narrow-band optical filter subsystem **4** eliminating wide band spectral interference; a light-sensing device (e.g. photo-diode) **56** mounted at the focal point of the light collection mirror **55**, for detecting the filtered narrow-band optical signal focused therein by the light collecting mirror **55**; and an electronic circuitry **57** for processing electrical signals produced by the photo-diode **56** indicative of the intensity of detected light exposure levels within the focal plane of the CMOS image sensing array **22**. During light exposure measurement operations, incident narrow-band LED-based illumination is gathered from the center of the FOV of the system by the spherical light collecting mirror **55** and narrow-band filtered by the narrow-band optical filter subsystem **4** before being focused upon the photodiode **56** for intensity detection. The photo-diode **56** converts the detected light signal into an electrical signal having an amplitude which directly corresponds to the intensity of the collected light signal.

As shown in FIG. **7A1** of an alternative illustrative embodiment, the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** comprises: a mirrored beam splitter **55**A mounted within the head portion of the hand-supportable housing for collecting narrow-band LED-based light reflected from a central portion of the FOV of the system, wherein image light is reflected off the object to be imaged and upon the mirrored beam splitter **55**A resulting in a portion of the image light being transmitted through the mirrored beam splitter **55**A and focused upon a photodiode **56** for subsequent processing.

The mirrored beam splitter in its most common form is an optical device that splits a beam of light in two. The mirrored beam splitter comprises a plate of glass with a thin coating of silver (usually deposited from silver vapour) with the thickness of the silver coated such that of light incident at a 45 degree angle, one portion is transmitted and one portion is reflected. The relative percentages of the reflection/transmission ratio may vary. Instead of a silver coating, a dielectric optical coating may be used instead.

As a general principal, it is desirable to gather light from the center of the camera's FOV, which is where the object being imaged is most likely to be positioned. Using the mirrored beam splitter set up as described above, the optical axes of the image formation and detection subsystem and the photodiode may be made exactly coincident, with the extra advantage of enabling one dimension of the scanner to be reduced. In alternative embodiments the photodiode may have a lens or lenses in front of it to aid in light collection.

As shown in FIG. **7A2** of an alternative illustrative embodiment, the Automatic Light Exposure Measurement and Illumination Control Subsystem **15** comprises: a cube-type beam splitter **55**A mounted within the head portion of the hand-supportable housing for collecting narrow-band LED-based light reflected from a central portion of the FOV of the system, wherein image light is reflected off the object to be imaged and upon the cube-type beam splitter **55**A resulting in a portion of the image light being transmitted through the cube-type beam splitter **55**A and focused upon a photodiode **56** for subsequent processing.

The cube-type beam splitter in its most common form is an optical device that splits a beam of light in two. The cube-type beam splitter comprises two triangular glass prisms which are

glued together at their base. The thickness of the resin layer is adjusted such that (for a certain wavelength) a portion of the light incident through one face of the cube is reflected and the other portion is transmitted. The relative percentages of the reflection/transmission ratio may vary.

As a general principal, it is desirable to gather light from the center of the camera's FOV, which is where the object being imaged is most likely to be positioned. Using the cube-type beam splitter set up as described above, the optical axes of the image formation and detection subsystem and the photodiode may be made exactly coincident, with the extra advantage of enabling one dimension of the scanner to be reduced. In alternative embodiments the photodiode may have a lens or lenses in front of it to aid in light collection.

As shown in FIG. **7**B, the System Control Subsystem **19** generates an Illumination Array Selection Control Signal which determines which LED illumination array (i.e. the narrow-area illumination array **27** or the far-field and narrow-field wide-area illumination arrays **28** or **29**) will be selectively driven at any instant in time of system operation by LED Array Driver Circuitry **64** in the Automatic Light Exposure Measurement and Illumination Control Subsystem **15**. As shown, electronic circuitry **57** processes the electrical signal from photo-detector **56** and generates an Auto-Exposure Control Signal for the selected LED illumination array. In term, this Auto-Exposure Control Signal is provided to the LED Array Driver Circuitry **64**, along with an Illumination Array Selection Control Signal from the System Control Subsystem **19**, for selecting and driving (i.e. energizing) one or more LED illumination array(s) so as to generate visible illumination at a suitable intensity level and for suitable time duration so that the CMOS image sensing array **22** automatically detects digital high-resolution images of illuminated objects, with sufficient contrast and brightness, while achieving Global Exposure Control objectives of the present invention disclosed herein. As shown in FIGS. **7**B and **7**C, the Illumination Array Selection Control Signal is generated by the System Control Subsystem **19** in response to (i) reading the System Mode Configuration Parameters from the System Mode Configuration Parameter Table **70**, shown in FIG. **2**A**1**, for the programmed mode of system operation at hand, and (ii) detecting the output from the automatic IR-based Object Presence and Range Detection Subsystem **12**.

Notably, in the illustrative embodiment, there are three possible LED-based illumination arrays **27**, **28** and **29** which can be selected for activation by the System Control Subsystem **19**, and the upper and/or lower LED subarrays in illumination arrays **28** and **29** can be selectively activated or deactivated on a subarray-by-subarray basis, for various purposes taught herein, including automatic specular reflection noise reduction during wide-area image capture modes of operation.

Each one of these illumination arrays can be driven to different states depending on the Auto-Exposure Control Signal generated by electronic signal processing circuit **57**, which will be generally a function of object distance, object surface reflectivity and the ambient light conditions sensed at photo-detector **56**, and measured by signal processing circuit **57**. The operation of signal processing circuitry **57** will now be detailed below.

As shown in FIG. **7**B, the narrow-band filtered optical signal that is produced by the parabolic light focusing mirror **55** is focused onto the photo-detector D**1** **56** which generates an analog electrical signal whose amplitude corresponds to the intensity of the detected optical signal. This analog electrical signal is supplied to the signal processing circuit **57** for various stages of processing. The first step of processing involves converting the analog electrical signal from a current-based signal to a voltage-based signal which is achieved by passing it through a constant-current source buffer circuit **58**, realized by one half of transistor Q**1** (**58**). This inverted voltage signal is then buffered by the second half of the transistor Q**1** (**58**) and is supplied as a first input to a summing junction **59**. As shown in FIG. **7**C, the CMOS image sensing array **22** produces, as output, a digital Electronic Rolling Shutter (ERS) pulse signal **60**, wherein the duration of this ERS pulse signal **60** is fixed to a maximum exposure time allowed in the system. The ERS pulse signal **60** is buffered through transistor Q**2** **61** and forms the other side of the summing junction **59**. The outputs from transistors Q**1** and Q**2** form an input to the summing junction **59**. A capacitor C**5** is provided on the output of the summing junction **59** and provides a minimum integration time sufficient to reduce any voltage overshoot in the signal processing circuit **57**. The output signal across the capacitor C**5** is further processed by a comparator U**1** **62**. In the illustrative embodiment, the comparator reference voltage signal is set to 1.7 volts. This reference voltage signal sets the minimum threshold level for the light exposure measurement circuit **57**. The output signal from the comparator **62** is inverted by inverter U**3** **63** to provide a positive logic pulse signal which is supplied, as Auto-Exposure Control Signal, to the input of the LED array driver circuit **64** shown in FIG. **7**C.

As will be explained in greater detail below, the LED Array Driver Circuit **64** shown in FIG. **7**C automatically drives an activated LED illuminated array, and the operation of LED Array Driver Circuit **64** depends on the mode of operation in which the Multi-Mode Illumination Subsystem **14** is configured. In turn, the mode of operation in which the Multi-Mode Illumination Subsystem **14** is configured at any moment in time will typically depend on (i) the state of operation of the Object Presence and Range Detection Subsystem **12** and (ii) the programmed mode of operation in which the entire Imaging-Based Bar Code Symbol Reading System is configured using System Mode Configuration Parameters read from the Table **70** shown in FIG. **2**A**1**.

As shown in FIG. **7**C, the LED Array Driver Circuit **64** comprises analog and digital circuitry which receives two input signals: (i) the Auto-Exposure Control Signal from signal processing circuit **57**; and (ii) the Illumination Array Selection Control Signal. The LED Array Driver Circuit **64** generates, as output, digital pulse-width modulated (PCM) drive signals provided to either the narrow-area illumination array **27**, the upper and/or lower LED subarray employed in the near-field wide-area illumination array **28**, and/or the upper and/or lower LED subarrays employed in the far-field wide-area illumination array **29**. Depending on which Mode of System Operation the Imaging-Based Bar Code Symbol Reader has been configured, the LED Array Driver Circuit **64** will drive one or more of the above-described LED illumination arrays during object illumination and imaging operations. As will be described in greater detail below, when all rows of pixels in the CMOS image sensing array **22** are in a state of integration (and thus have a common integration time), such LED illumination array(s) are automatically driven by the LED Array Driver Circuit **64** at an intensity and for duration computed (in an analog manner) by the Automatic Light Exposure and Illumination Control Subsystem **15** so as to capture digital images having good contrast and brightness, independent of the light intensity of the ambient environment and the relative motion of target object with respect to the Imaging-Based Bar Code Symbol Reader.

Global Exposure Control Method of the Present Invention Carried Out Using the CMOS Image Sensing Array

In the illustrative embodiment, the CMOS image sensing array **22** is operated in its Single Frame Shutter Mode (i.e. rather than its Continuous Frame Shutter Mode) as shown in FIG. **7**D, and employs a novel exposure control method which ensure that all rows of pixels in the CMOS image sensing array **22** have a common integration time, thereby capturing high quality images even when the object is in a state of high speed motion. This novel exposure control technique shall be referred to as "the global exposure control method" of the present invention, and the flow chart of FIGS. **7**E**1** and **7**E**2** describes clearly and in great detail how this method is implemented in the Imaging-Based Bar Code Symbol Reader of the illustrative embodiment. The global exposure control method will now be described in detail below

As indicated at Block A in FIG. **7**E**1**, Step A in the global exposure control method involves selecting the single frame shutter mode of operation for the CMOS imaging sensing array provided within an imaging-based bar code symbol reading system employing an automatic light exposure measurement and illumination control subsystem, a multi-mode illumination subsystem, and a system control subsystem integrated therewith, and image formation optics providing the CMOS image sensing array with a field of view into a region of space where objects to be imaged are presented.

As indicated in Block B in FIG. **7**E**1**, Step B in the global exposure control method involves using the automatic light exposure measurement and illumination control subsystem to continuously collect illumination from a portion of the field of view, detect the intensity of the collected illumination, and generate an electrical analog signal corresponding to the detected intensity, for processing.

As indicated in Block C in FIG. **7**E**1**, Step C in the global exposure control method involves activating (e.g. by way of the system control subsystem **19** or directly by way of trigger switch **2**C) the CMOS image sensing array so that its rows of pixels begin to integrate photonically generated electrical charge in response to the formation of an image onto the CMOS image sensing array by the image formation optics of the system.

As indicated in Block D in FIG. **7**E**1**, Step D in the global exposure control method involves the CMOS image sensing array **22** automatically (i) generating an Electronic Rolling Shutter (ERS) digital pulse signal when all rows of pixels in the image sensing array are operated in a state of integration, and providing this ERS pulse signal to the Automatic Light Exposure Measurement And Illumination Control Subsystem **15** so as to activate light exposure measurement and illumination control functions/operations there within.

As indicated in Block E in FIG. **7**E**2**, Step E in the global exposure control method involves, upon activation of light exposure measurement and illumination control functions within Subsystem **15**, (i) processing the electrical analog signal being continuously generated there within, (ii) measuring the light exposure level within a central portion of the field of view **23** (determined by light collecting optics **55** shown in FIG. **7**A), and (iii) generating an Auto-Exposure Control Signal for controlling the generation of visible field of illumination from at least one LED-based illumination array (**27**, **28** and/or **29**) in the Multi-Mode Illumination Subsystem **14** which is selected by an Illumination Array Selection Control Signal produced by the System Control Subsystem **19**.

Finally, as indicated at Block F in FIG. **7**E**2**, Step F in the global exposure control method involves using (i) the Auto-Exposure Control Signal and (ii) the Illumination Array Selection Control Signal to drive the selected LED-based illumination array(s) and illuminate the field of view of the CMOS image sensing array **22** in whatever image capture mode it may be configured, precisely when all rows of pixels in the CMOS image sensing array are in a state of integration, as illustrated in FIG. **7**D, thereby ensuring that all rows of pixels in the CMOS image sensing array have a common integration time. By enabling all rows of pixels in the CMOS image sensing array **22** to have a common integration time, high-speed "global exposure control" is effectively achieved within the Imaging-Based Bar Code Symbol Reader of the present invention, and consequently, high quality images are captured independent of the relative motion between the Bar Code Symbol Reader and the target object.

Specification of the IR-Based Automatic Object Presence and Range Detection Subsystem Employed in the Hand-Supportable Digital Image-Based Bar Code Reading Device of the Present Invention

As shown in FIG. **8**A, IR-wavelength based Automatic Object Presence and Range Detection Subsystem **12** is realized in the form of a compact optics module **76** mounted on the front portion of optics bench **6**, as shown in FIG. **1**J.

As shown in FIG. **8**, the Object Presence and Range Detection Module **12** of the illustrative embodiment comprises a number of subcomponents, namely: an optical bench **77** having an ultra-small footprint for supporting optical and electro-optical components used to implement the subsystem **12**; at least one IR laser diode **78** mounted on the optical bench **77**, for producing a low power IR laser beam **79**; IR beam shaping optics **80**, supported on the optical bench for shaping the IR laser beam (e.g. into a pencil-beam like geometry) and directing the same into the central portion of the object detection field **20** defined by the field of view (FOV) of IR light collection/focusing optics **81** supported on the optical bench **77**; an amplitude modulation (AM) circuit **82** supported on the optical bench **77**, for modulating the amplitude of the IR laser beam produced from the IR laser diode at a frequency f_0 (e.g. 75 Mhz) with up to 7.5 milliwatts of optical power; optical detector (e.g. an avalanche-type IR photodetector) **83**, mounted at the focal point of the IR light collection/focusing optics **81**, for receiving the IR optical signal reflected off an object within the object detection field, and converting the received optical signal **84** into an electrical signal **85**; an amplifier and filter circuit **86**, mounted on the optical bench **77**, for isolating the f_0 signal component and amplifying it; a limiting amplifier **87**, mounted on the optical bench, for maintaining a stable signal level; a phase detector **88**, mounted on the optical bench **77**, for mixing the reference signal component f_0 from the AM circuit **82** and the received signal component f_0 reflected from the packages and producing a resulting signal which is equal to a DC voltage proportional to the Cosine of the phase difference between the reference and the reflected f_0 signals; an amplifier circuit **89**, mounted on the optical bench **77**, for amplifying the phase difference signal; a received signal strength indicator (RSSI) **90**, mounted on the optical bench **77**, for producing a voltage proportional to a LOG of the signal reflected from the target object which can be used to provide additional information; a reflectance level threshold analog multiplexer **91** for rejecting information from the weak signals; and a 12 bit A/D converter **92**, mounted on the optical bench **77**, for converting the DC voltage signal from the RSSI circuit **90** into sequence of time-based range data elements $\{R_{n,i}\}$, taken along nT discrete instances in time, where each range data element $R_{n,i}$ provides a measure of the distance of the object referenced from (i) the IR laser diode **78** to (ii) a point on the surface of

the object within the object detection field **20**; and Range Analysis Circuitry **93** described below.

In general, the function of Range Analysis Circuitry **93** is to analyze the digital range data from the A/D converter **90** and generate two control activation signals, namely: (i) "an object presence detection" type of control activation signal A_{1A} indicating simply whether an object is presence or absent from the object detection field, regardless of the mode of operation in which the Multi-Mode Illumination Subsystem **14** might be configured; and (ii) "a near-field/far-field" range indication type of control activation signal A_{1B} indicating whether a detected object is located in either the predefined near-field or far-field portions of the object detection field, which correspond to the near-field and far-field portions of the FOV of the Multi-Mode Image Formation and Detection Subsystem **13**.

Various kinds of analog and digital circuitry can be designed to implement the IR-based Automatic Object Presence and Range Detection Subsystem **12**. Alternatively, this subsystem can be realized using various kinds of range detection techniques as taught in U.S. Pat. No. 6,637,659, incorporated herein by reference in its entirely.

In the illustrative embodiment, Automatic Object Presence and Range Detection Subsystem **12** operates as follows. In System Modes of Operation requiring automatic object presence and/or range detection, Automatic Object Presence and Range Detection Subsystem **12** will be activated at system start-up and operational at all times of system operation, typically continuously providing the System Control Subsystem **19** with information about the state of objects within both the far and near portions of the object detection field **20** of the Imaging-Based Symbol Reader. In general, this Subsystem detects two basic states of presence and range, and therefore has two basic states of operation. In its first state of operation, the IR-based automatic Object Presence and Range Detection Subsystem **12** automatically detects an object within the near-field region of the FOV **20**, and in response thereto generates a first control activation signal which is supplied to the System Control Subsystem **19** to indicate the occurrence of this first fact. In its second state of operation, the IR-based automatic Object Presence and Range Detection Subsystem **12** automatically detects an object within the far-field region of the FOV **20**, and in response thereto generates a second control activation signal which is supplied to the System Control Subsystem **19** to indicate the occurrence of this second fact. As will be described in greater detail and throughout this Patent Specification, these control activation signals are used by the System Control Subsystem **19** during particular stages of the system control process, such as determining (i) whether to activate either the near-field and/or far-field LED illumination arrays, and (ii) how strongly should these LED illumination arrays be driven to ensure quality image exposure at the CMOS image sensing array **22**.

Specification of the Mapping of Pixel Data Captured by the Imaging Array into the SDRAM Under the Control of the Direct Memory Access (DMA) Module within the Microprocessor

As shown in FIG. **9**, the CMOS image sensing array **22** employed in the Digital Imaging-Based Bar Code Symbol Reading Device hereof is operably connected to its microprocessor **36** through FIFO **39** (realized by way of a FPGA) and system bus shown in FIG. **2**B. As shown, SDRAM **38** is also operably connected to the microprocessor **36** by way of the system bus, thereby enabling the mapping of pixel data captured by the CMOS image sensing array **22** into the SDRAM **38** under the control of the direct memory access (DMA) module within the microprocessor **36**.

Referring to FIG. **10**, details will now be given on how the bytes of pixel data captured by CMOS image sensing array **22** are automatically mapped (i.e. captured and stored) into the addressable memory storage locations of its SDRAM **38** during each image capture cycle carried out within the hand-supportable imaging-based bar code reading device of the present invention.

In the implementation of the illustrative embodiment, the CMOS image sensing array **22** sends 8-bit gray-scale data bytes over a parallel data connection to FPGA **39** which implements a FIFO using its internal SRAM. The FIFO **39** stores the pixel data temporarily and the microprocessor **36** initiates a DMA transfer from the FIFO (which is mapped to address OXOCOOOOOO, chip select 3) to the SDRAM **38**. In general, modern microprocessors have internal DMA modules, and a preferred microprocessor design, the DMA module will contain a 32-byte buffer. Without consuming any CPU cycles, the DMA module can be programmed to read data from the FIFO **39**, store read data bytes in the DMA's buffer, and subsequently write the data to the SDRAM **38**. Alternatively, a DMA module can reside in FPGA **39** to directly write the FIFO data into the SDRAM **38**. This is done by sending a bus request signal to the microprocessor **36**, so that the microprocessor **36** releases control of the bus to the FPGA **39** which then takes over the bus and writes data into the SDRAM **38**.

Below, a brief description will be given on where pixel data output from the CMOS image sensing array **22** is stored in the SDRAM **38**, and how the microprocessor (i.e. implementing a decode algorithm) **36** accesses such stored pixel data bytes. FIG. **10** represents the memory space of the SDRAM **38**. A reserved memory space of 1.3 MB is used to store the output of the CMOS image sensing array **22**. This memory space is a 1:1 mapping of the pixel data from the CMOS image sensing array **22**. Each byte represents a pixel in the image sensing array **22**. Memory space is a mirror image of the pixel data from the image sensing array **22**. Thus, when the decode program (**36**) accesses the memory, it is as if it is accessing the raw pixel image of the image sensing array **22**. No time code is needed to track the data since the modes of operation of the bar code reader guarantee that the microprocessor **36** is always accessing the up-to-date data, and the pixel data sets are a true representation of the last optical exposure. To prevent data corruption, i.e. new data coming in while old data are still being processed, the reserved space is protected by disabling further DMA access once a whole frame of pixel data is written into memory. The DMA module is re-enabled until either the microprocessor **36** has finished going through its memory, or a timeout has occurred.

During image acquisition operations, the image pixels are sequentially read out of the image sensing array **22**. Although one may choose to read and column-wise or row-wise for some CMOS image sensors, without loss of generality, the row-by-row read out of the data is preferred. The pixel image data set is arranged in the SDRAM **38** sequentially, starting at address OXAOEC0000. To randomly access any pixel in the SDRAM **38** is a straightforward matter: the pixel at row y ¼ column x located is at address (OXAOEC0000+y×1280+x).

As each image frame always has a frame start signal out of the image sensing array **22**, that signal can be used to start the DMA process at address OXAOEC0000, and the address is continuously incremented for the rest of the frame. But the reading of each image frame is started at address OXAOEC0000 to avoid any misalignment of data. Notably, however, if the microprocessor **36** has programmed the

CMOS image sensing array **22** to have a ROI window, then the starting address will be modified to (OXAOEC0000+1280×R_1), where R_1 is the row number of the top left corner of the ROI.

Specification of the Three-Tier Software Architecture of the Hand-Supportable Digital Image-Based Bar Code Reading Device of the Present Invention

As shown in FIG. **11**, the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention **1** is provided with a three-tier software architecture comprising the following software modules: (1) the Main Task module, the CodeGate Task module, the Metroset Task module, the Application Events Manager module, the User Commands Table module, and the Command Handler module, each residing within the Application layer of the software architecture; (2) the Tasks Manager module, the Events Dispatcher module, the Input/Output Manager module, the User Commands Manager module, the Timer Subsystem module, the Input/Output Subsystem module and the Memory Control Subsystem module, each residing within the System Core (SCORE) layer of the software architecture; and (3) the Linux Kemal module, the Linux File System module, and Device Drivers modules, each residing within the Linux Operating System (OS) layer of the software architecture.

While the operating system layer of the Imaging-Based Bar Code Symbol Reader is based upon the Linux operating system, it is understood that other operating systems can be used (e.g. Microsoft Windows, Max OXS, Unix, etc), and that the design preferably provides for independence between the main Application Software Layer and the Operating System Layer, and therefore, enables of the Application Software Layer to be potentially transported to other platforms. Moreover, the system design principles of the present invention provides an extensibility of the system to other future products with extensive usage of the common software components, which should make the design of such products easier, decrease their development time, and ensure their robustness.

In the illustrative embodiment, the above features are achieved through the implementation of an event-driven multi-tasking, potentially multi-user, Application layer running on top of the System Core software layer, called SCORE. The SCORE layer is statically linked with the product Application software, and therefore, runs in the Application Level or layer of the system. The SCORE layer provides a set of services to the Application in such a way that the Application would not need to know the details of the underlying operating system, although all operating system APIs are, of course, available to the application as well. The SCORE software layer provides a real-time, event-driven, OS-independent framework for the product Application to operate. The event-driven architecture is achieved by creating a means for detecting events (usually, but not necessarily, when the hardware interrupts occur) and posting the events to the Application for processing in real-time manner. The event detection and posting is provided by the SCORE software layer. The SCORE layer also provides the product Application with a means for starting and canceling the software tasks, which can be running concurrently, hence, the multi-tasking nature of the software system of the present invention.

Specification of Software Modules within the Score Layer of the System Software Architecture Employed in Imaging-Based Bar Code Reader of the Present Invention

The SCORE layer provides a number of services to the Application layer.

The Tasks Manager provides a means for executing and canceling specific application tasks (threads) at any time during the product Application run.

The Events Dispatcher provides a means for signaling and delivering all kinds of internal and external synchronous and asynchronous events

When events occur, synchronously or asynchronously to the Application, the Events Dispatcher dispatches them to the Application Events Manager, which acts on the events accordingly as required by the Application based on its current state. For example, based on the particular event and current state of the application, the Application Events Manager can decide to start a new task, or stop currently running task, or do something else, or do nothing and completely ignore the event.

The Input/Output Manager provides a means for monitoring activities of input/output devices and signaling appropriate events to the Application when such activities are detected.

The Input/Output Manager software module runs in the background and monitors activities of external devices and user connections, and signals appropriate events to the Application Layer, which such activities are detected. The Input/Output Manager is a high-priority thread that runs in parallel with the Application and reacts to the input/output signals coming asynchronously from the hardware devices, such as serial port, user trigger switch **2**C, bar code reader, network connections, etc. Based on these signals and optional input/output requests (or lack thereof) from the Application, it generates appropriate system events, which are delivered through the Events Dispatcher to the Application Events Manager as quickly as possible as described above.

The User Commands Manager provides a means for managing user commands, and utilizes the User Commands Table provided by the Application, and executes appropriate User Command Handler based on the data entered by the user.

The Input/Output Subsystem software module provides a means for creating and deleting input/output connections and communicating with external systems and devices The Timer Subsystem provides a means of creating, deleting, and utilizing all kinds of logical timers.

The Memory Control Subsystem provides an interface for managing the multi-level dynamic memory with the device, fully compatible with standard dynamic memory management functions, as well as a means for buffering collected data. The Memory Control Subsystem provides a means for thread-level management of dynamic memory. The interfaces of the Memory Control Subsystem are fully compatible with standard C memory management functions. The system software architecture is designed to provide connectivity of the device to potentially multiple users, which may have different levels of authority to operate with the device.

The User Commands Manager, which provides a standard way of entering user commands, and executing application modules responsible for handling the same. Each user command described in the User Commands Table is a task that can be launched by the User Commands Manager per user input, but only if the particular user's authority matches the command's level of security.

The Events Dispatcher software module provides a means of signaling and delivering events to the Application Events Manager, including the starting of a new task, stopping a currently running task, or doing something or nothing and simply ignoring the event.

FIG. **12**B provides a Table listing examples of System-Defined Events which can occur and be dispatched within the hand-supportable Digital Imaging-Based Bar Code Symbol

Reading Device of the present invention, namely: SCORE_EVENT_POWER_UP which signals the completion of system start-up and involves no parameters; _ SCORE_EVENT_TIMEOUT which signals the timeout of the logical timer, and involves the parameter "pointer to timer id"; SCORE_EVENT_UNEXPECTED_INPUT which signals that the unexpected input data is available and involves the parameter "pointer to connection id"; SCORE_EVENT_TRIG_ON which signals that the user pulled the trigger and involves no parameters; SCORE_EVENT_TRIG_OFF which signals that the user released the trigger and involves no parameters; SCORE_EVENT_OBJECT_DETECT_ON which signals that the object is positioned under the bar code reader and involves no parameters; SCORE_EVENT_OBJECT_DETECT_OFF which signals that the object is removed from the field of view of the bar code reader and involves no parameters; SCORE_EVENT_EXIT_TASK which signals the end of the task execution and involves the pointer UTID; and SCORE_EVENT_ABORT_TASK which signals the aborting of a task during execution.

The Imaging-Based Bar Code Symbol Reader of the present invention provides the user with a command-line interface (CLI), which can work over the standard communication lines, such as RS232, available in the Bar Code Reader. The CLI is used mostly for diagnostic purposes, but can also be used for configuration purposes in addition to the MetroSet® and MetroSelect® programming functionalities. To send commands to the bar code reader utilizing the CLI, a user must first enter the User Command Manager by typing in a special character, which could actually be a combination of multiple and simultaneous keystrokes, such Ctrl and S for example. Any standard and widely available software communication tool, such as Windows HyperTerminal, can be used to communicate with the Bar Code Reader. The bar code reader acknowledges the readiness to accept commands by sending the prompt, such as "MTLG>" back to the user. The user can now type in any valid Application command. To quit the User Command Manager and return the scanner back to its normal operation, a user must enter another special character, which could actually be a combination of multiple and simultaneous keystrokes, such Ctrl and R for example.

An example of the valid command could be the "Save Image" command, which is used to upload an image from the bar code reader's memory to the host PC. This command has the following CLI format:

save [filename [compr]]

where

(1) save is the command name.

(2) filename is the name of the file the image gets saved in. If omitted, the default filename is "image.bmp".

(3) compr is the compression number, from 0 to 10. If omitted, the default compression number is 0, meaning no compression. The higher compression number, the higher image compression ratio, the faster image transmission, but more distorted the image gets.

The Imaging-Based Bar Code Symbol Reader of the present invention can have numerous commands. All commands are described in a single table (User Commands Table shown in FIG. **11**) contained in the product Applications software layer. For each valid command, the appropriate record in the table contains the command name, a short description of the command, the command type, and the address of the function that implements the command.

When a user enters a command, the User Command Manager looks for the command in the table. If found, it executes the function the address of which is provided in the record for the entered command. Upon return from the function, the User Command Manager sends the prompt to the user indicating that the command has been completed and the User Command Manager is ready to accept a new command.

Specification of Software Modules within the Application Layer of the System Software Architecture Employed in Imaging-Based Bar Code Reader of the Present Invention

The image processing software employed within the system hereof performs its bar code reading function by locating and recognizing the bar codes within the frame of a captured image comprising pixel data. The modular design of the image processing software provides a rich set of image processing functions, which could be utilized in the future for other potential applications, related or not related to bar code symbol reading, such as: optical character recognition (OCR) and verification (OCV); reading and verifying directly marked symbols on various surfaces; facial recognition and other biometrics identification; etc.

The CodeGate Task, in an infinite loop, performs the following task. It illuminates a "thin" narrow horizontal area at the center of the field-of-view (FOV) and acquires a digital image of that area. It then attempts to read bar code symbols represented in the captured frame of image data using the image processing software facilities supported by the Image-Processing Bar Code Symbol Reading Subsystem **17** of the present invention to be described in greater detail hereinafter. If a bar code symbol is successfully read, then Subsystem **17** saves the decoded data in the special CodeGate data buffer. Otherwise, it clears the CodeGate data buffer. Then, it continues the loop. The CodeGate Task never exits on its own. It can be canceled by other modules in the system when reacting to other events. For example, when a user pulls the trigger switch **2**C, the event TRIGGER_ON is posted to the application. The Application software responsible for processing this event, checks if the CodeGate Task is running, and if so, it cancels it and then starts the Main Task. The CodeGate Task can also be canceled upon OBJECT_DETECT_OFF event, posted when the user moves the bar code reader away from the object, or when the user moves the object away from the bar code reader.

Depending on the System Mode in which the Imaging-Based Bar Code Symbol Reader is configured, Main Task will typically perform differently. For example, when the Imaging-Based Bar Code Symbol Reader is configured in the Programmable Mode of System Operation No. 12 (i.e. Semi-Automatic-Triggered Multiple-Attempt 1D/2D Single-Read Mode) to be described in greater detail hereinafter, the Main Task first checks if the CodeGate Data Buffer contains data decoded by the CodeGate Task. If so, then it immediately sends the data out to the user by executing the Data Output procedure and exits. Otherwise, in a loop, the Main Task does the following: it illuminates an entire area of the field-of-view and acquires a full-frame image of that area. It attempts to read a bar code symbol the captured image. If it successfully reads a bar code symbol, then it immediately sends the data out to the user by executing the Data Output procedure and exits. Otherwise, it continues the loop. Notably, upon successful read and prior to executing the Data Output procedure, the Main Task analyzes the decoded data for a "reader programming" command or a sequence of commands. If necessary, it executes the MetroSelect functionality. The Main Task can be canceled by other modules within the system when reacting to other events. For example, the bar code reader of the present invention can be re-configured using standard Metrologic configuration methods, such as

MetroSelect® and MetroSet®. The MetroSelect functionality is executed during the Main Task.

The MetroSet functionality is executed by the special MetroSet Task. When the Focus RS232 software driver detects a special NULL-signal on its communication lines, it posts the METROSET_ON event to the Application. The Application software responsible for processing this event starts the MetroSet task. Once the MetroSet Task is completed, the scanner returns to its normal operation.

Operating System Layer Software Modules within the Application Layer of the System Software Architecture Employed in Imaging-Based Bar Code Reader of the Present Invention

The Devices Drivers software modules, which includes trigger drivers, provides a means for establishing a software connection with the hardware-based manually-actuated trigger switch **2**C employed on the imaging-based device, an image acquisition driver for implementing image acquisition functionality aboard the imaging-based device, and an IR driver for implementing object detection functionality aboard the imaging-based device.

As shown in FIG. **12**I, the Device Drive software modules include: trigger drivers for establishing a software connection with the hardware-based manually-actuated trigger switch **2**C employed on the Imaging-Based Bar Code Symbol Reader of the present invention; an image acquisition driver for implementing image acquisition functionality aboard the Imaging-Based Bar Code Symbol Reader; and an IR driver for implementing object detection functionality aboard the Imaging-Based Bar Code Symbol Reader.

Basic System Operations Supported by the Three-Tier Software Architecture of the Hand-Supportable Digital Imaging-Based Bar Code Reading Device of the Present Invention

In FIGS. **13**A through **13**L, the basic systems operations supported by the three-tier software architecture of the digital Imaging-Based Bar Code Reading Device of the present invention are schematically depicted. Notably, these basic operations represent functional modules (or building blocks) with the system architecture of the present invention, which can be combined in various combinations to implement the numerous Programmable Modes of System Operation listed in FIG. **23** and described in detail below, using the image acquisition and processing platform disclosed herein. For purposes of illustration, and the avoidance of obfuscation of the present invention, these basic system operations will be described below with reference to Programmable Mode of System Operation No. 12: Semi-Automatic-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing The No-Finder Mode And The Manual Or Automatic Modes Of the Multi-Mode Bar Code Reading Subsystem **17**.

FIG. **13**A shows the basic operations carried out within the System Core Layer of the system when the user points the bar code reader towards a bar code symbol on an object. Such operations include the by IR device drivers enabling automatic detection of the object within the field, and waking up of the Input/Output Manager software module. As shown in FIG. **13**B, the Input/Output Manager then posts the SCORE_OBJECT_DETECT_ON event to the Events Dispatcher software module in response to detecting an object. Then as shown in FIG. **13**C, the Events Dispatcher software module passes the SCORE_OBJECT_DETECT_ON event to the Application Layer. FIG. **13**D shows that, upon receiving the SCORE_OBJECT_DETECT_ON event at the Application Layer, the Application Events Manager executes an event handling routine (shown in FIG. **13**D) which activates the narrow-area (linear) illumination array **27** (i.e. during narrow-area illumination and image capture modes), and then executes the CodeGate Task described in FIG. **13**E. As shown in the flow chart of FIG. **13**D, the system event handling routine first involves determining whether the CodeGate Task has been enabled and if not, then the system exits the event handling routine. If the CodeGate Task has been enabled (as would be the case for Programmable Mode of System Operation No. 12, in particular), then the routine proceeds to determine whether the "Presentation Mode" (i.e. Programmable Mode of System Operation No. 10) has been enabled, and if so, then the Application Layer executes the Main Task shown in FIG. **13**J, and if not, then activates the narrow area illumination mode of the Multi-Mode Illumination Subsystem **14**, and then executes the CodeGate Task shown in FIG. **13**E.

As shown in FIG. **13**E, the Application Layer executes the CodeGate Task by first activating the narrow-area image capture mode in the Multi-Mode Image Formation and Detection Subsystem **13** (i.e. by enabling a few middle rows of pixels in the CMOS sensor array **22**), and then acquiring/capturing a narrow image at the center of the FOV of the Bar Code Reader. CodeGate Task then performs image processing operations on the captured narrow-area image using No-Finder Module which has been enabled by the selected Programmable Mode of System Operation No. 12. If the image processing method results in a successful read of a bar code symbol, then the Codegate Task saves the decoded symbol character data in the Codegate Data Buffer; and if not, then the task clears the Codegate Data Buffer, and then returns to the main block of the Task where image acquisition reoccurs.

As shown in FIG. **13**F, when the user pulls the trigger switch **2**C on the bar code reader while the Code Task is executing, the trigger switch driver in the OS Layer automatically wakes up the Input/Output Manager at the System Core Layer. As shown in FIG. **13**G, the Input/Output Manager, in response to being woken up by the trigger device driver, posts the SCORE_TRIGGER_ON event to the Events Dispatcher also in the System Core Layer. As shown in FIG. **13**H, the Events Dispatcher then passes on the SCORE_TRIGGER_ON event to the Application Events Manager at the Application Layer. As shown in FIG. **13**I, the Application Events Manager responds to the SCORE_TRIGGER_ON event by invoking a handling routine (Trigger On Event) within the Task Manager at the System Core Layer. As shown the flow chart of FIG. **13**I, the routine determines whether the Presentation Mode (i.e. Programmed Mode of System Operation No. 10) has been enabled, and if so, then the routine exits. If the routine determines that the Presentation Mode (i.e. Programmed Mode of System Operation No. 10) has not been enabled, then it determines whether the CodeGate Task is running, and if not, then, executes the Main Task described in FIG. **13**J. If the routine determines that the CodeGate Task is running, then it first cancels the CodeGate Task and then deactivates the narrow-area illumination array **27** associated with the Multi-Mode Illumination Subsystem **14**, and thereafter executes the Main Task.

As shown in FIG. **13**J, the first step performed in the Main Task by the Application Layer is to determine whether CodeGate Data is currently available (i.e. stored in the CodeGate Data Buffer), and if such data is available, then the Main Task directly executes the Data Output Procedure described in FIG. **13**K. However, if the Main Task determines that no such data is currently available, then it starts the Read TimeOut Timer, and then acquires a wide-area image of the detected object, within the time frame permitted by the Read Timeout Timer. Notably, this wide-area image acquisition process involves carrying out the following operations, namely: (i) first activating the wide-area illumination mode in the Multi-

Mode Illumination Subsystem **14** and the wide-area capture mode in the CMOS image formation and detection module; (ii) determining whether the object resides in the near-field or far-field portion of the FOV (through object range measurement by the IR-based Object Presence and Range Detection Subsystem **12**); and (iii) then activating either the near or far field wide-area illumination array to illuminate either the object in either the near or far field portions of the FOV using either the near-field illumination array **28** or the far-field illumination array **29** (or possibly both **28** and **29** in special programmed cases) at an intensity and duration determined by the automatic light exposure measurement and control subsystem **15**; while (iv) sensing the spatial intensity of light imaged onto the CMOS image sensing array **22** in accordance with the Global Exposure Control Method of the present invention, described in detail hereinabove. Then the Main Task performs image processing operations on the captured image using either the Manual, ROI-Specific or Automatic Modes of operation, although it is understood that other image-processing based reading methods taught herein, such as Automatic or OmniScan, can be used depending on which Programmed Mode of System Operation has been selected by the end user for the Imaging-Based Bar Code Symbol Reader of the present invention. Notably, in the illustrative embodiment shown in FIG. **13**J, the time duration of each image acquisition/processing frame is set by the Start Read Timeout Timer and Stop Read Timeout Timer blocks shown therein, and that within the Programmed Mode of System Operation No. 12, the Main Task will support repeated (i.e. multiple) attempts to read a single bar code symbol so long as the trigger switch **2**C is manually depressed by the operator and a single bar code has not yet been read. Then upon successfully reading a (single) bar code symbol, the Main Task will then execute the Data Output Procedure. Notably, in other Programmed Modes of System Operation, in which a single attempt at reading a bar code symbol is enabled, the Main Task will be modified accordingly to support such system behavior. In such a case, an alternatively named Main Task (e.g. Main Task No. 2) would be executed to enable the required system behavior during run-time.

It should also be pointed out at this juncture, that it is possible to enable and utilize several of different kinds of symbol reading methods during the Main Task, and to apply particular reading methods based on the computational results obtained while processing the narrow-area image during the CodeGate Task, and/or while preprocessing of the captured wide-area image during one of the image acquiring/processing frames or cycles running in the Main Task. The main point to be made here is that the selection and application of image-processing based bar code reading methods will preferably occur through the selective activation of the different modes available within the multi-mode image-processing based bar code symbol reading Subsystem **17**, in response to information learned about the graphical intelligence represented within the structure of the captured image, and that such dynamic should occur in accordance with principles of dynamic adaptive learning commonly used in advanced image processing systems, speech understanding systems, and alike. This general approach is in marked contrast with the approaches used in prior art Imaging-Based Bar Code Symbol Readers, wherein permitted methods of bar code reading are pre-selected based on statically defined modes selected by the end user, and not in response to detected conditions discovered in captured images on a real-time basis.

As shown in FIG. **13**K, the first step carried out by the Data Output Procedure, called in the Main Task, involves determining whether the symbol character data generated by the Main Task is for programming the bar code reader or not. If the data is not for programming the Bar Code Symbol Reader, then the Data Output Procedure sends the data out according to the bar code reader system configuration, and then generates the appropriate visual and audio indication to the operator, and then exits the procedure. If the data is for programming the Bar Code Symbol Reader, then the Data Output Procedure sets the appropriate elements of the bar code reader configuration (file) structure, and then saves the Bar Code Reader Configuration Parameters in non-volatile RAM (i.e. NOVRAM). The Data Output Procedure then reconfigures the Bar Code Symbol Reader and then generates the appropriate visual and audio indication to the operator, and then exits the procedure. As shown in FIG. **13**L, decoded data is sent from the Input/Output Module at the System Core Layer to the Device Drivers within the Linux OS Layer of the system.

Wide-Area Illumination Control Method for Use During the Main Task System Control Routine So as to Illuminate Objects with Wide-Area Illumination in a Manner which Substantially Reduces Specular-Type Reflection at the CMOS Image Sensing Array of the Bar Code Symbol Reader

Referring to FIGS. **13M1** through **13M3**, the method of illuminating objects without specular reflection, according to the present invention, will now be described in detail. This control routine can be called during the acquisition of wide-area image step in the Main Task routine, shown in FIG. **13**J.

As indicated at Step A in FIG. **13M1**, the first step of the illumination control method involves using the Automatic Light Exposure Measurement And Illumination Control Subsystem **15** to measure the ambient light level to which the CMOS image sensing array **22** is exposed prior to commencing each illumination and imaging cycle within the Bar Code Symbol Reading System

As indicated at Step B, the illumination control method involves using the Automatic IR-based Object Presence and Range Detection Subsystem **12** to measure the presence and range of the object in either the near or far field portion of the field of view (FOV) of the System.

As indicated at Step C, the illumination control method involves using the detected range and the measured light exposure level to drive both the upper and lower LED illumination subarrays associated with either the near-field wide-area illumination array **28** or far-field wide-area illumination array **29**.

As indicated at Step D, the illumination control method involves capturing a wide-area image at the CMOS image sensing array **22** using the illumination field produced during Step C.

As indicated at Step E, the illumination control method involves rapidly processing the captured wide-area image during Step D to detect the occurrence of high spatial-intensity levels in the captured wide-area image, indicative of a specular reflection condition.

As indicated at Step F, the illumination control method involves determining if a specular reflection condition is detected in the processed wide-area image, and if so then driving only the upper LED illumination subarray associated with either the near-field or far-field wide-area illumination array. Also, if a specular reflection condition is not detected in the processed wide-area image, then the detected range and the measured light exposure level is used to drive both the upper and lower LED subarrays associated with either the near-field or far-field wide-area illumination array.

As indicated at Step G, the illumination control method involves capturing a wide-area image at the CMOS image sensing array **22** using the illumination field produced during Step F.

As indicated at Step H, the illumination control method involves rapidly processing the captured wide-area image during Step G to detect the occurrence of high spatial-intensity levels in the captured wide-area image, indicative of a specular reflection condition.

As indicated at Step I, the illumination control method involves determining if a specular reflection condition is still detected in the processed wide-area image, and if so, then drive the other LED subarray associated with either the near-field or far-field wide-area illumination array. If a specular reflection condition is not detected in the processed wide-area image, then the detected Range and the measured Light Exposure Level is used to drive the same LED illumination subarray (as in Step C) associated with either the near-field wide-area illumination array **28** or far field wide-area illumination array **29**.

As indicated at Step J, the illumination control method involves capturing a wide-area image at the CMOS image sensing array using the illumination field produced during Step I.

As indicated at Step K, the illumination control method involves rapidly processing the captured wide-area image during Step J to detect the absence of high spatial-intensity levels in the captured wide-area image, confirming the elimination of the earlier detected specular reflection condition.

As indicated at Step L, the illumination control method involves determining if no specular reflection condition is detected in the processed wide-area image at Step K, and if not, then the wide-area image is processed using the mode(s) selected for the Multi-Mode Image-Processing Bar Code Reading Subsystem **17**. If a specular reflection condition is still detected in the processed wide-area image, then the control process returns to Step A repeats Steps A through K, as described above.

Specification of Symbologies and Modes Supported by the Multi-Mode Bar Code Symbol Reading Subsystem Module Employed within the Hand-Supportable Digital Image-Based Bar Code Reading Device of the Present Invention

FIG. **14** lists the various bar code symbologies supported by the Multi-Mode Bar Code Symbol Reading Subsystem **17** employed within the hand-supportable Digital Imaging-Based Bar Code Symbol Reading Device of the present invention. As shown therein, these bar code symbologies include: Code 128; Code 39; I2of5; Code93; Codabar; UPC/EAN; Telepen; UK-Plessey; Trioptic; Matrix 2of5; Ariline 2of5; Straight 2of5; MSI-Plessey; Code11; and PDF417.

Specification of the Various Modes of Operation in the Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention

As shown in FIG. **15**, the Multi-Mode Image-Processing Based Bar Code Symbol Reading Subsystem **17** of the illustrative embodiment supports five primary modes of operation, namely: the Automatic Mode of Operation; the Manual Mode of Operation; the ROI-Specific Mode of Operation; the No-Finder Mode of Operation; and Omniscan Mode of Operation. As will be described in greater detail herein, various combinations of these modes of operation can be used during the lifecycle of the image-processing based bar code reading process of the present invention.

FIG. **16** is a exemplary flow chart representation showing the steps involved in setting up and cleaning up the software sub-Application entitled "Multi-Mode Image-Processing Based Bar Code Symbol Reading Subsystem **17**", once called from either (i) the CodeGate Task software module at the Block entitled READ BAR CODE(S) **1**N CAPTURED NARROW-AREA IMAGE indicated in FIG. **13**E, or (ii) the Main Task software module at the Block entitled "READ BAR CODE(S) IN CAPTURED WIDE-AREA IMAGE" indicated in FIG. **13**J.

The Automatic Mode of Multi-Mode Bar Code Symbol Reading Subsystem

In its Automatic Mode of Operation, the Multi-Mode Bar Code Symbol Reading Subsystem **17** is configured to automatically start processing a captured frame of digital image data, prior to the complete buffering thereof, so as to search for one or more bar codes represented therein in an incremental manner, and to continue searching until the entire image is processed.

This mode of image-based processing enables bar code locating and reading when no prior knowledge about the location of, or the orientation of, or the number of bar codes that may be present within an image, is available. In this mode of operation, the Multi-Mode Bar Code Symbol Reading Subsystem **17** starts processing the image from the top-left corner and continues until it reaches the bottom-right corner, reading any potential bar codes as it encounters them.

The Manual Mode of the Multi-Mode Bar Code Symbol Reading Subsystem

In its Manual Mode of Operation, the Multi-Mode Bar Code Symbol Reading Subsystem **17** is configured to automatically process a captured frame of digital image data, starting from the center or sweep spot of the image at which the user would have aimed the bar code reader, so as to search for (i.e. find) a at least one bar code symbol represented therein. Unlike the Automatic Mode, this is done by searching in a helical manner through frames or blocks of extracted image feature data, and then marking the same and image-processing the corresponding raw digital image data until a bar code symbol is recognized/read within the captured frame of image data.

This mode of image processing enables bar code locating and reading when the maximum number of bar codes that could be present within the image is known a priori and when portions of the primary bar code have a high probability of spatial location close to the center of the image. The Multi-Mode Bar Code Symbol Reading Subsystem **17** starts processing the image from the center, along rectangular strips progressively further from the center and continues until either the entire image has been processed or the programmed maximum number of bar codes has been read.

The ROI-Specific Mode of the Multi-Mode Bar Code Symbol Reading Subsystem

In its ROI-Specific Mode of Operation, the Multi-Mode Bar Code Symbol Reading Subsystem **17** is configured to automatically process a captured frame of digital image data, starting from the region of interest (ROI) in the captured image, specified by coordinates acquired during a previous mode of operation within the Multi-Mode Bar Code Symbol Reading Subsystem **17**. Unlike the Manual Mode, this is done by analyzing the received ROI-specified coordinates, derived during either, a previous NoFinder Mode, Automatic Mode, or Omniscan Mode of operation, and then immediately begins processing image feature data, and image-processing the corresponding raw digital image data until a bar code symbol is recognized/read within the captured frame of image data. Thus, typically, the ROI-Specific Mode is used in

conjunction with other modes of the Multi-Mode Bar Code Symbol Reading Subsystem **17**.

This mode of image processing enables bar code locating and reading when the maximum number of bar codes that could be present within the image is known a priori and when portions of the primary bar code have a high probability of spatial location close to specified ROI in the image. The Multi-Mode Bar Code Symbol Reading Subsystem starts processing the image from these initially specified image coordinates, and then progressively further in a helical manner from the ROI-specified region, and continues until either the entire image has been processed or the programmed maximum number of bar codes have been read.

The No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem

In its No-Finder Mode of Operation, the Multi-Mode Bar Code Symbol Reading Subsystem **17** is configured to automatically process a captured narrow-area (linear) frame of digital image data, without the feature extraction and marking operations used in the Automatic, Manual and ROI-Specific Modes, so as to read a one or more bar code symbols represented therein.

This mode enables bar code reading when it is known, a priori, that the image contains at most one (1-dimensional) bar code symbol, portions of which have a high likelihood of spatial location close to the center of the image and when the bar code is known to be oriented at zero degrees relative to the horizontal axis. Notably, this is typically the case when the bar code reader is used in a hand-held mode of operation, where the Bar Code Symbol Reader is manually pointed at the bar code symbol to be read. In this mode, the Multi-Mode Bar Code Symbol Reading Subsystem **17** starts at the center of the image, skips all bar code location steps, and filters the image at zero (0) degrees and 180 degrees relative to the horizontal axis. Using the "bar-and-space-count" data generated by the filtration step, it reads the potential bar code symbol.

The Omni-Scan Mode of the Multi-Mode Bar Code Reading Subsystem

In its Omniscan Mode of Operation, the Multi-Mode Bar Code Symbol Reading Subsystem **17** is configured to automatically process a captured frame of digital image data along any one or more predetermined virtual scan line orientations, without feature extraction and marking operations used in the Automatic, Manual and ROI-Specific Modes, so as to read a single bar code symbol represented in the processed image.

This mode enables bar code reading when it is known, a priori, that the image contains at most one (1-dimensional) bar code, portions of which have a high likelihood of spatial location close to the center of the image but which could be oriented in any direction. Multi-Mode Bar Code Symbol Reading Subsystem **17** starts at the center of the image, skips all bar code location steps, and filters the image at different start-pixel positions and at different scan-angles. Using the bar-and-space-count data generated by the filtration step, the Omniscan Mode reads the potential bar code symbol.

Specification of Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its Automatic Mode of Operation

As shown in FIG. **17**A, the image-processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its Automatic Mode of operation, comprises the following primary steps of operation, namely: (1) the first stage of processing involves searching for (i.e. finding) regions of interest (ROIs) by processing a low resolution image of a captured frame of high-resolution image data, partitioning the low-resolution image into N×N blocks, creating a feature vector (Fv) for each block using spatial-derivative based image processing techniques, marking ROIs by examining the feature vectors for regions of high-modulation, (2) the second stage of processing involves calculating bar code orientation, and marking the four corners of a bar code as a ROI, and (3) the third stage of processing involves reading any bar code symbols represented within the ROI by traversing the bar code image data, updating the feature vectors, examining the zero-crossings of filtered image data, creating bar and space patterns, and decoding the bar and space patterns using conventional decoding algorithms.

As will be described herein below, these three (3) stages of image processing involved in the Automatic Mode of operation can be sub-divided into four major processing blocks (i.e. modules), namely: the Tracker Module **100**, the Finder Module **101**, the Marker Module **102**, and the Decoder Module **103**, which are shown in FIG. **2A2** and described in detail below. When the Automatic Mode of the Multi-Mode Bar Code Symbol Reading Subsystem **17** is invoked, these four processing blocks (i.e. modules) are executed, sequentially, and optionally incrementally so that a rectangular sub-region of the entire image can be processed per invocation.

First Stage of Image-Based Processing within the Multi-Mode Bar Code Symbol Reading Subsystem During its Automatic Mode of Operation

During its Automatic Mode of operation, the first stage of processing in the Multi-Mode Bar Code Symbol Reading Subsystem **17** comprises: (i) searching for (i.e. finding) regions of interest (ROIs) by processing a low resolution image of a captured frame of high-resolution image data as shown in FIG. **18**A; (ii) partitioning the low-resolution image of the package label into N×N blocks as shown in FIG. **18**B; (iii) creating a feature vector for each block of low-resolution image data as shown in FIG. **18**C using gradient vectors, edge density measures, the number of parallel edge vectors, centroids of edgels, intensity variance, and the histogram of intensities captured from the low-resolution image; (iv) examining the feature vectors for regions for parallel lines by detection of high modulation, high edge density, large number of parallel edge vectors and large intensity variance (using spatial-derivative based image processing techniques) as shown in FIG. **18**D; and (v) marking ROIs. In general, this stage of processing is started before all lines of the full digital image data frame are buffered in memory, and typically only requires the number of rows in a given (first) feature block to be buffered in memory before the reading process can begin.

Detailed Specification of the Tracker Module

As indicated at Blocks A, B, C, C**1** and XX in FIG. **17**B, the first invocation of the Tracker Module **100** resets the Finder Module **101**, Marker Module **102**, and Decoder Module **103** sub-components to their initial state (as Block A); it resets the feature vector array Fv (at Block D) and the number of Regions of Interest (ROI). All subsequent invocations set the maximum processing line number of each of the three blocks to the current y-dimension of the image. The Tracker Module invokes an optional callback function (Pause Checker) to facilitate aborting or pausing Multi-Mode Bar Code Symbol Reading Subsystem **17** or to change parameters on the fly.

Detailed Specification of the Finder Module

As indicated at Blocks D through Y in FIG. **17**B, the Finder Module **101** (processing block) sub-divides the image into N×N blocks, each of which has a feature vector array (Fv) element associated with it. An Fv element contains a set of

numbers that identify the strong possibility of the presence of parallel lines within that image block. As indicated at Blocks D through Y, the Finder Module **101** processes the image at a lower spatial resolution; it processes every n^{th} line and every n^{th} pixel within each of the selected lines thereby performing calculations on the original image down-sampled-by-n. For each selected line it calculates:

$$\overline{I_y} = \frac{n \sum_{x=1}^{N_x} I(x, y)}{N_x} \tag{1}$$

where I(x, y)=gray value at pixel location (x, y) and

N_x=x-dimension of the supplied (sub)image

If $\overline{I}_y$ by exceeds a programmable "background threshold", the image line y is declared a foreground line and is processed further by the Finder Module. A pixel is declared as a background pixel if its gray value is below a certain threshold. The Finder Module starts from the left-most pixel and traverses right on the foreground line, finds at Block G the first pixel whose intensity (gray value) exceeds the programmable background threshold and marks it as the left-edge (x_l) of the line. At Block H, the Finder Module then starts from the right-most pixel and traversing leftward on the foreground line determines the right-edge (x_r) using the same method. For foreground line y the Finder Module calculates at Block I:

$$I'_1(x, y)=|I(x+1, y)-I(x-1, y)|+|I(x, y+11)-I(x, y-1)|, \text{ where } x_l \leqq x \leqq x_r \tag{2}$$

If $I'_l(x, y)$ exceeds a threshold at Block J, the Finder Module marks pixel (x,y) as an edge element or edgel.

In order to find the direction and magnitude of the edge-vector corresponding to edgel (x,y), the Finder Module calculates at Block K:

$$(I')_0(x, y) = \left| \begin{array}{l} w_1^0 I(x-1, y-1) + w_2^0 I(x, y-1) + \\ w_3^0 I(x+1, y-1) + w_4^0 I(x-1, y) + w_5^0 I(x, y) + \\ w_6^0 I(x+1, y) + w_7^0 I(x-1, y+1) + \\ w_8^0 I(x, y+1) + w_9^0 I(x+1, y+1) \end{array} \right| \tag{3}$$

$$(I')_{45}(x, y) = \left| \begin{array}{l} w_1^{45} I(x-1, y-1) + w_2^{45} I(x, y-1) + \\ w_3^{45} I(x+1, y-1) + w_4^{45} I(x-1, y) + w_5^{45} I(x, y) + \\ w_6^{45} I(x+1, y) + w_7^{45} I(x-1, y+1) + \\ w_8^{45} I(x, y+1) + w_9^{45} I(x+1, y+1) \end{array} \right| \tag{4}$$

$$(I')_{90}(x, y) = \left| \begin{array}{l} w_1^{90} I(x-1, y-1) + w_2^{90} I(x, y-1) + \\ w_3^{90} I(x+1, y-1) + w_4^{90} I(x-1, y) + w_5^{90} I(x, y) + \\ w_6^{90} I(x+1, y) + w_7^{90} I(x-1, y+1) + \\ w_8^{90} I(x, y+1) + w_9^{90} I(x+1, y+1) \end{array} \right| \tag{5}$$

$$(I')_{135}(x, y) = \left| \begin{array}{l} w_1^{135} I(x-1, y-1) + w_2^{135} I(x, y-1) + \\ w_3^{135} I(x+1, y-1) + w_4^{135} I(x-1, y) + \\ w_5^{135} I(x, y) + w_6^{135} I(x+1, y) + \\ w_7^{135} I(x-1, y+1) + w_8^{135} I(x, y+1) + \\ w_9^{135} I(x+1, y+1) \end{array} \right| \tag{6}$$

where the coefficients $w_i^0, w_i^{45}, w_i^{90}, w_i^{135}$ are given by the operators:

$$w^0 = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \quad w^{45} = \begin{matrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{matrix} \quad w^{90} = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

$$w^{135} = \begin{matrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{matrix}$$

At Block M, the Finder Module updates the Fv block that edgel (x,y) belongs to with:

$$\text{Edge strength: } I'_{fv_i} = \sum_{j=1}^{n} I'_{ij} \tag{7}$$

$$\text{where } I'_{ij} = \text{edge strength of edgel } j\text{: and}$$

$$n = \text{number of edgels inside } Fv \text{ block } i$$

$$\text{Edge direction: } A_{fv_i}(z) = \sum_{j=1}^{n} A_j, \text{ where } A_j = \begin{cases} 1, & j = k, k \in [0, 3] \\ 0 \end{cases} \quad I'_{z_1} \geq I'_{z_2} \geq I'_{z_3} \geq I'_{z_4}, z_i = 45 * (k + i - 1) \tag{8}$$

$$\text{Centroid of edgels: } \overline{x}_{fv_i} = \frac{\sum_{j=1}^{n} x_j}{n}, \overline{y}_{fv_i} = \frac{\sum_{j=1}^{n} y_j}{n} \text{ where } (x_j, y_j) \text{ are the coordinates of edgels} \tag{9}$$

$$\text{Cumulative histogram: } H_{fv_i}(z) = \sum_{j=1}^{n} H_j, \text{ where } H_j = \begin{cases} 1, & I(x, y) \leq z \\ 0 \end{cases} \tag{10}$$

At Block N, the Finder Module goes through all the lines of the current image section and populates the Fv array using the above-mentioned features. At Blocks O through U, the Finder Module checks to see if all lines have been processed.

At Block V, the Finder Module then examines each Fv array element for features that strongly point to the presence of parallel lines within the Fv block. At Block W, an interesting Fv is declared as part of a Region of Interest (ROI) when the number of edgels exceeds a threshold, at least one of the edgel direction array elements exceeds a threshold value, and

$$m-n>C, \text{ where}$$

$$H_{fv_i}(m) > \alpha N, H_{fv_i}(n) > (1-\alpha)N, \tag{11}$$

C=Contrast-threshold

$\alpha \in (0,1)$

N=total number of pixels in image block corresponding to feature vector array Fv Notably, at Blocks C, E, and T, the Finder Module invokes the Pause Checker callback function to let the scanning application take control.

Second Stage of Image-Based Processing within the Multi-Mode Bar Code Symbol Reading Subsystem During its Automatic Mode of Operation

During its Automatic Mode of Operation, the second stage of processing in the Multi-Mode Bar Code Symbol Reading

Subsystem **17** involves (ii) calculating bar code orientation by analyzing the feature vectors for parallel lines, and (ii) marking the four corners of a bar code as a ROI, in terms of xy coordinates.

FIGS. **18**E and **18**F illustrate calculating bar code orientation, during the second marking stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem **17** during its Automatic Mode of operation, wherein within each feature vector block, the scan line data representing the bar code is traversed (i.e. sliced) at different angles, the slices are matched with each other based on "least mean square error", and the correct orientation is determined to be that angle which matches the mean square error sense through every slice of the bar code.

FIG. **18**G illustrates the marking of the four corners of the detected bar code symbol, during the second marking stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem **17** during its Automatic Mode of operation. During this stage of processing, such marking operations are performed on the full high-resolution image of the parcel, the bar code is traversed in either direction starting from the center of the block, the extent of modulation is detected using the intensity variance, and the x,y coordinates (pixels) of the four corners of the bar code are detected starting from 1 and 2 and moving perpendicular to the bar code orientation, so as to ultimately define the ROI by the detected four corners of the bar code symbol within the high-resolution image.

Detailed Specification of the Marker Module

Within the Multi-Mode Bar Code Symbol Reading Subsystem **17** shown in FIG. **2**A**2**, the Marker Module as indicated at Blocks Z through KK, in FIG. **17**B, takes over from the Finder Module and examines each ROI to determine the complete extent of the ROI. The Finder Module then checks the location of the centroid of the ROI and compares it to the line number of the accumulated images in memory.

$$y_{roi_i}+L>N_y \tag{12}$$

where

y_{roi_i}=y coordinate of the centroid of ROI_i

L=Maximum length (in pixels) of any bar code presented to Multi-Mode Bar Code Symbol Reading Subsystem

N_y=y-dimension of cumulative image

If inequality (12) holds, then the Marker Module postpones calculations for this ROI until the y-dimension of the image is such that inequality does not hold. When the Marker Module continues to process the ROI, it first determines the orientation of the parallel lines that could potentially be part of a bar code, by calculating:

$$\theta = \left(225 - \tan^{-1}\left(\frac{I'_{135}}{I'_{45}}\right)\right)\mathrm{mod}(180),\ I'_0 \geq I'_{45},\ I'_0 \geq I'_{45},\ I'_0 \geq I'_{135} \tag{13}$$

$$\theta = \left(\tan^{-1}\left(\frac{I'_{90}}{I'_0}\right)\right)\mathrm{mod}(180),\ I'_{45} \geq I'_0,\ I'_{45} \geq I'_{90},\ I'_{45} \geq I'_{135}$$

$$\theta = \left(45 + \tan^{-1}\left(\frac{I'_{135}}{I'_{45}}\right)\right)\mathrm{mod}(180),$$
$$I'_{90} \geq I'_{45},\ I'_{90} \geq I'_0,\ I'_{90} \geq I'_{135}$$

$$\theta = \left(180 - \tan^{-1}\left(\frac{I'_{90}}{I'_0}\right)\right)\mathrm{mod}(180),$$
$$I'_{135} \geq I'_0,\ I'_{135} \geq I'_{90},\ I'_{135} \geq I'_{45}$$

$$\begin{bmatrix} x_{j+1} \\ y_{j+1} \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} \tag{14}$$

-continued

$$\begin{bmatrix} x'_j \\ y'_j \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} - n_i \begin{bmatrix} -\sin\beta \\ \cos\beta \end{bmatrix} \tag{15}$$

The angle θ that yields the minimum E(β), is assumed to be a close approximation of the actual orientation angle of the parallel lines.

Having calculated the correct orientation of the parallel lines, the Marker Module calculates the narrowest and the widest width of the parallel lines in the neighborhood of the ROI by traversing (i.e. scanning) the image in the direction of orientation of the lines as well as at 180 degrees to it (e.g. using a spot size window of say N×N pixels (e.g. where 1<N<10). It should be noted that all angle measurements are clockwise relative to the horizontal axis. Equation (14) specifies the traversal equation with β=θ,θ+180. Details of the method used to calculate the widths of the lines are explained at length in the Decoder Module section.

The Marker Module uses the widths of the narrowest and widest elements to determine a pixel count (n) that closely approximates the minimum quiet-zone allowable for any bar code symbology. It then traverses the image again using equation (14) and calculates:

$$m_i = \frac{\sum_{j=i}^{i+n} I(x_j, y_j)}{n} \tag{16}$$

$$v_i = \frac{\sum_{j=i}^{i+n} \|I(x_j, y_j) - m_i\|}{n-1}$$

$$IV_i = \frac{v_i}{m_i^2}$$

where m_i=mean of the set of n pixels starting at pixel i

v_i=variance of the set of n pixels starting at pixel I

If IV_i is less than a threshold, then the Marker Module makes the assumption that the group of parallel lines end at pixel i (similarly for the θ+180 direction). Starting from pixel i and traversing the image using (15) and a spot size window of say N×N pixels (e.g. where 1<N<10), and performing similar calculations as in equation (16) the four corners that approximate the quadrilateral bound of the potential bar code are determined. A pictorial representation of the above-mentioned method can be found in the figure entitled "Step 6: Mark ROIs: Mark four corners of bar code.

The Marker Module then marks all the Fv blocks that encompass the quadrilateral bound of the potential bar code, with the current ROI identifier; if there already exists one or more ROIs with different identifiers, the Marker Module picks that ROI that completely encompasses the others. The old ROIs are kept only if they are not completely enclosed within the current ROI.

The Marker Module also frequently invokes the Pause Checker to let the bar code reading Application (running) take over control.

Third Stage of Image-Based Processing within the Multi-Mode Bar Code Symbol Reading Subsystem During its Automatic Mode of Operation

The third stage of processing involves reading any bar code symbols represented within the ROI by traversing the bar code and updating the feature vectors, examining the zero-

crossings of filtered images, creating bar and space patterns, and decoding the bar and space patterns

FIG. **18**H shows updating the feature vectors during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem during its Automatic Mode of operation. During this stage of processing, the histogram component of the feature vector Fv is updated while traversing the bar code (using a spot size window of say N×N pixels (e.g. where $1<N<10$), the estimate of the black-to-white transition is calculated, and an estimate of narrow and wide elements of the bar code are also calculated.

FIG. **18**I illustrates the search for zero crossings during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem **17** during its Automatic Mode of operation. During this stage of processing, the high-resolution bar code image is median filtered in a direction perpendicular to bar code orientation, the second derivative zero crossings define edge crossings, the zero-crossing data is used only for detecting edge transitions, and the Black/White transition estimates are used to put upper and lower bounds to bar and space grey levels, as graphically illustrated.

FIG. **18**J illustrates creating a bar and space pattern during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem during its Automatic Mode of operation. During this stage of processing, the edge transition is modeled as a ramp function, the edge transition is assumed to be 1 pixel wide, the edge transition location is determined at the sub-pixel level, and the bar and space counts are gathered using edge transition data;

FIG. **18**K illustrates generating the decode bar and space pattern during the third stage of processing within the Multi-Mode Bar Code Symbol Reading Subsystem during its Automatic Mode of operation. During this stage of operation, the bar and space data is framed with borders, and the bar and space data is decoded using existing laser scanning bar code decoding algorithms.

Detailed Specification of the Decoder Module

As indicated in at Blocks LL through AAA in **17**B, the Decoder Module takes over from the Marker Module and examines each ROI previously defined by the Marker Module. For each ROI, the Decoder Module uses the quadrilateral bound coordinates {x,y} to calculate the longer (higher) extremity of the potential bar code (towards the possible quiet-zones). The Decoder Module then computes the maximum number of possible scan-lines as:

$$T = \frac{D}{n} \tag{17}$$

where D=length of the longer extremity, and n=pixel-offset per scan-line.

Notably, the parameter n (i.e. pixel-offset per scan line) represents how far the Decoder Module moves up its virtual scan direction (parallel to the previous virtual scan direction) and processes the image during each image processing cycle. As any captured image will be corrupted by some degree of noise (and certainly greater levels when a bar code symbol cannot be decoded), the Decoder Module needs to perform its next processing cycle on a line of scan data that is located as far away as possible from the previous line of scan data which did not result in a successful decode, but at the same time, the Decoder Module should exploit the inherent noise-immunity features provided in many bar code symbologies. Thus, in accordance with the present invention, the pixel-offset per scan line variable n is not arbitrarily selected, as in most prior art systems, but rather is determined by carefully (i) determining the maximum pixel height (length) of the ROI under consideration, and (ii) dividing this maximum pixel height of the ROI into a number of pixel-offset distances proportional to the maximum pixel height of the ROI. In the preferred embodiment, the number or sequence of scan lines into which the ROI can be divided for subsequent cycles of image processing, thus defining the pixel off-set per scan-line, is described by the formula: $f(m, n)=(2m-1)/2^{n-1}$, where $n=1,2\ldots N$. and $1<m<2^{n-1}$.

The Decoder Module traverses the potential bar code using equation (14) and calculates approximations for the first and second order derivatives:

$$I_i' = \sum_{j=-1}^{1} \begin{bmatrix} w_1 I(x_j-1, y_j-1) + w_2 I(x_j, y_j-1) + \\ w_3 I(x_j+1, y_j-1) + w_4 I(x_j-1, y_j) + \\ w_5 I(x_j, y_j) + w_6 I(x_j+1, y_j) + \\ w_7 I(x_j-1, y_j+1) + w_8 I(x_j, y_j+1) + \\ w_9 I(x_j+1, y_j+1) \end{bmatrix} \tag{18}$$

$$I_i'' = I_{i+1}' - I_{i-1}'$$

where

$$w_i = \begin{matrix} 0.776 & 0.000 & -0.776 \\ 1.000 & 0.000 & -1.000 \\ 0.776 & 0.000 & -0.776 \end{matrix} \ldots 0 < \theta \le 22 \tag{19}$$

$$w_i = \begin{matrix} 1.000 & 0.776 & 0.000 \\ 0.776 & 0.000 & -0.776 \\ 0.000 & -0.776 & -1.000 \end{matrix} \ldots 0 < \theta \le 68$$

$$w_i = \begin{matrix} 0.776 & 1.000 & 0.776 \\ 0.000 & 0.000 & 0.000 \\ -0.776 & -1.000 & -0.776 \end{matrix} \ldots 0 < \theta \le 113$$

$$w_i = \begin{matrix} 0.000 & 0.776 & 1.000 \\ -0.776 & 0.000 & 0.776 \\ -1.000 & -0.776 & 0.000 \end{matrix} \ldots 0 < \theta \le 158$$

$$w_i = \begin{matrix} -0.776 & 0.000 & 0.776 \\ -1.000 & 0.000 & 1.000 \\ -0.776 & 0.000 & 0.776 \end{matrix} \ldots 158 < \theta < 180$$

and (x_j, y_j) are related by equation (15).

The Decoder Module examines the zero crossings of I''_i and if

$$I''_i \cdot I''_{i+1} > 0, \text{ and}$$

$$I''_{i+1} < 0, \text{ and}$$

$$I'_i > -T \tag{20}$$

where T=minimum derivative magnitude threshold, then the Decoder Module concludes that a "space to bar transition" has occurred.

If:

$$I''_i \cdot I''_{i+1} < 0, \text{ and}$$

$$I'_{i+1} < 0, \text{ and}$$

$$I'_i > T \tag{21}$$

then, the Decoder Module concludes that a "bar to space transition" has occurred.

The Decoder Module takes the difference in pixel position of adjacent bar/space transitions and adds it to the interpolated mid-point of the bar-space/space-bar transition (found

using I'_i) to determine the width of each element of the potential bar code. This is the same method used by the Marker Module to calculate the widths of the narrowest and widest parallel lines.

Having calculated the "bar-and-space-count" data for each scan-line, the Decoder Module invokes the different (and separately enabled) symbology-decoders supported within the Imaging-Based Bar Code Symbol Reader, as indicated at FIG. **18**K. Each symbology decoder, whether 1-dimensional or certain 2-dimensional symbologies (like PDF417), detects the presence of the correct number of bars and spaces and also the correct start/stop pattern before attempting to decode the potential bar code symbol.

If the Decoder Module decodes using the current "scan-line data", then it skips all other scan-lines. If the Decoder Module detects a stacked symbology, then it continues to gather more scan-line-data. If decoding fails, then the Decoder Module adjusts the scan-line angles (bar code-orientation angle) progressively and repeats the process. The Decoder Module, in the process of collecting scan-line-data, also correlates the bar-and-space-data from one scan-line with that of the adjacent scan-lines in order to read through damaged or poorly presented bar codes. For every bar code that is decoded by the Decoder Module, a callback function is invoked to save the decoded result. The Decoder Module calls the Pause Checker callback function frequently to let the scanning application take control.

In its Automatic Mode, the Multi-Mode Bar Code Symbol Reading Subsystem **17** repeats this entire process for the entire image, and optionally for progressively acquired images.

Specification of Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its Manual Mode of Operation

FIG. **19**A illustrates the steps involved in the process carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its Manual Mode of operation. During this manual mode of operation, the first stage of processing involves searching for and finding regions of interest (ROIs) by processing a low resolution image of a captured frame of high-resolution image data, partitioning the low-resolution image into N×N blocks, and creating a feature vector for the middle block using spatial-derivative based image processing techniques. Then the second stage of processing involves marking ROIs by examining the feature vectors for regions of high-modulation and returning to the first stage to create feature vectors for other blocks surrounding the middle block (in a helical manner), calculating bar code orientation and eventually marking the four corners of a bar code as a ROI, and (3) the third stage of processing involves reading any bar code symbols represented within the ROI by traversing the bar code and updating the feature vectors, examining the zero-crossings of filtered images, creating bar and space patterns, and decoding the bar and space patterns.

Like in the Automatic Mode, these three (3) stages of image processing in the manual mode of operation can be sub-divided into four major processing blocks (i.e. modules), namely: the Tracker Module, the Finder Module, the Marker Module, and the Decoder Module, which have been described in great detail above. When the Manual Mode of the Multi-Mode Bar Code Symbol Reading Subsystem **17** is invoked, these four processing blocks (i.e. modules) are executed sequentially and optionally incrementally so that a rectangular sub-region of the entire image can be processed per invocation.

FIG. **19**B illustrates the steps involved in the decode process carried out by the Multi-Mode Bar Code Symbol Reading Subsystem **17** during its Manual Mode of operation. As indicated at Block A, the Main Task or CodeGate Task in Application Layer invokes the Tracker Module to find the center coordinates of the center block of captured image data, to which the center feature vector will be associated. This central block of image data will be associated with image pixels located along the central portion of the image frame captured by the Imaging-Based Bar Code Symbol Reader. This step involves the Tracker Module resetting the Finder Module, Marker Module, and Decoder Module sub-components to their initial state; it resets the feature vector array and the number of Regions of Interest (ROI). While not indicated in the flow chart of FIG. **19**B, the Tracker Module invokes an optional callback function (Pause Checker) at various location within the control flow to facilitate aborting or pausing Multi-Mode Bar Code Symbol Reading Subsystem **17** or to change parameters on the fly.

As indicated at Block B in FIG. **19**B, the Finder Module is invoked and the captured image is subdivided into N×N blocks, each of which has a feature vector (Fv) array element associated with it. An Fv element contains a set of numbers that identify the strong possibility of the presence of parallel lines within that image block. As described hereinabove, the Finder Module processes the image at a lower spatial resolution; namely, it processes every n^{th} line and every n^{th} pixel within each of the selected lines thereby performing calculations on the original image down-sampled-by-n. For each selected line it calculates. At Block C, the Subsystem **17** determines if an ROI (bounding a complete bar code symbol) is found, and if so, invokes the Marker Module. Then at Block E, the Subsystem **17** determines whether an ROI has been marked by the Market Module, and if so, then the Decoder Module is invoked and then the ROI processed. If a bar code symbol is read within the ROI at Block G, then at Block H the Subsystem **17** determines if the actual number of decode cycles equals the required number of decode cycles. If so, then the Manual Mode of operation of the Subsystem **17** is stopped, and the flow returns to the Application Layer.

If at Block C in FIG. **19**B the Subsystem **17** determines that the ROI is not found, then the subsystem proceeds to Block I. If the Subsystem determines that all feature vectors have not yet been examined, then the Subsystem proceeds to Block J which advances the analysis to the next feature vector closet to the center feature vector, along the locus of a helical path through the image pixel data set. Then, at Block B, the Subsystem re-invokes the Finder Module to operate on this next feature vector.

If at Block G, the Subsystem determines that the Decoder Module does not successfully decode a bar code symbol in the ROI, then it advances to Block I and determines whether all feature vectors have not been examined.

The Subsystem **17** operated in the mode of operation specified by the flow chart of FIG. **19**B until a single bar code symbol is read within an ROI. Each instance of the Finder Module involves the analysis of another block of pixel data (corresponding to another feature vector) in effort to find an ROI containing a bar code symbol which can be found at Block C and successfully decoded at Block G. The sequential analysis of blocks of pixel data follows a helical pattern about the center starting point, determined at Block A of FIG. **19**B. Notably, during the Manual Mode of Operation, the Subsystem utilizes the image processing techniques described in connection with the Automatic Mode of operation, above.

The primary advantage of the Manual Mode of operation over the Automatic Mode of operation is that the Manual

Mode is that, when the user points the bar code reader at a bar code symbol to be read, the bar code reader in the manual mode is more likely to acquire an image and process the pixel data within a ROI containing a bar code symbol is a very quick manner, in comparison with the Automatic Mode which essentially scans and processes the entire captured image starting to from upper left most block of captured image data, ensuring a faster response time in hand-held bar code reading applications, in particular.

Specification of the Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its NoFinder Mode of Operation

FIG. **20**A illustrates that the image processing carried out by the Multi-Mode Bar Code Symbol Reading Subsystem **17** during its NoFinder Mode of operation involves essentially a single stage of image processing, unlike the Automatic, Manual and ROI-Specific Modes of operation. During this No-Finder Mode, Subsystem **17** does not employ the Tracker Module, the Finder Module or the Marker Module and instead only invokes the Decoder Module to (i) directly process the narrow-area high-resolution image captured by the bar code reader, one line of scan data at a time, starting from the middle thereof, (ii) examine the zero-crossings of the filtered image, (iii) create bar and space patterns therefrom, and then (iv) decode the bar and space patterns using conventional decoding algorithms. If the reading process is not successful, then the Subsystem **17** traverses another line of scan data within the captured narrow-area image, starting from a pixel offset n which is computed assuming a constant maximum height of the ROI which is deemed to be the pixel height of the captured narrow-area image.

FIG. **20**B illustrates the steps involved in the image processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem **17** during its NoFinder Mode of operation. As indicated at Block A in FIG. **20**B, the Subsystem **17** first finds (i.e. calculates) the center pixel in the captured narrow-area image. Then at Block B, the Subsystem **17** invokes the Decode Module and configures the same using the calculated center pixel. Within the Decode Module, sub-Blocks B**1** through B**8** are then carried out as shown in FIG. **20**A. As indicated in Block B**1**, the Decoder Module, starting from the calculated center point, scans the image horizontally and westward (using a spot-size window of say N×N pixels (e.g. where 1<N<10), and then processes the scanned image data to determine if a first border in a bar code symbol is found. Notably, this virtual scanning process is realized as a mathematical convolution of the spot-size window and the pixel data in the image buffer. If a first border is found at Block B**2**, then, once again starting from the calculated center point, the Decoder Module at Block B**3** scans the image horizontally and eastward (using a spot size window of say N×N pixels (e.g. where 1<N<10), and then at Block B**4** processes the scanned image data to determine if a second border in a bar code symbol is found. If a second border is found at Block B**4**, then the Decoder Module processes the captured image at Block B**5**. If, at Block B**6**, the Decoder Module successfully reads a bar code symbol within the scanned line of image data, then the Subsystem terminates the Decoder Module and stops the NoFinder Mode of operation.

If at Block B**2** in FIG. **20**A the Decoder Module does not find a first border of a bar code symbol, then it proceeds to Block B**7** and determines if it has tried all possible scan lines within the captured narrow-area image. If the Decoder Module has tried processing all possible scan lines through the narrow-area image, then it proceeds to the stop block and terminates the NoFinder Mode of operation. If the Decoder Module has not tried processing all scan lines through the captured narrow-area image, then it proceeds to Block B**8**, where it advances to the next line of scan data in the captured narrow-area image (i.e. by the offset pixel amount n), and then returns to Block B**1** where scanning and processing is resumed along the new scan line (using a spot size window of say N×N pixels (e.g. where 1<N<10).

If at Block B**4**, the second bar code border is not found, then the Decoder Module proceeds to Block B**7** and determines whether all scan lines through the captured image have been tried. If so, then the Subsystem **17** terminates the Decoder Module and exits its NoFinder Mode of operation. If all scan lines have not been tried at this stage of the process, then the Decoder Module proceeds to Block B**8** and advances to the next line of scan data for processing, as described hereinabove.

If at Block B**6** in FIG. **20**A the Decoder Module does not read a bar code within the current line of scan data being processed, then it proceeds to Block B**7**, where it determines if all lines of scan data have been tried. If all lines of scan data have not been tried, then at Block B**8** the Decoder Module advances to the next line of scan data in the captured narrow-area image (i.e. by the offset pixel amount n), and then returns to Block B**1** where scanning and processing is resumed along the new scan line (using a spot size window of say N×N pixels (e.g. where 1<N<10). If at Block B**7**, the Decoder Module determines that all lines of scan data have been tried, then the Decoder Module stops and terminates its process. For every bar code that is decoded by the Decoder Module, a callback function is invoked to save the decoded result. The Decoder Module calls the Pause Checker callback function frequently to let the car code symbol reading Application take control.

Specification of Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its Omniscan Mode of Operation

FIG. **21**A illustrates that the image processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its Omniscan Mode of operation involves essentially a single stage of image processing, unlike the Automatic, Manual and ROI-Specific Modes of operation. During this Omniscan Mode, the Decoder Module does not employ the Tracker Module, the Finder Mode or the Marker Module and instead directly processes the narrow-area high-resolution image captured by the bar code reader, along a plurality of spaced apart (e.g. 50 pixels) virtual scanning lines traversing through the entire 2D frame of image data captured by the Subsystem **17**. During the OmniScan Mode of operation, the Decoder Module assumes the imaged bar code symbol resides at the center of the captured wide-area high-resolution image with about a 1:1 aspect ratio (e.g. 1" tall×1" wide). Based on these assumptions, the Subsystem **17** starts at first predetermined angular orientation (e.g. 0, 30, 60, 90, 120 or 150 degrees), and then: (i) directly processes the high-resolution image along a set of parallel spaced-apart (e.g. 50 pixels) virtual scan lines line (using a spot size window of say N×N pixels (e.g. where 1<N<10); (ii) examines the zero-crossings along these virtual scan lines; (iii) creates bar and space patterns therefrom; and then (iv) decode processes the bar and space patterns. If image processing along the selected angular orientation fails to read a bar code symbol, then the Subsystem **17** automatically reprocesses the high-resolution image along a different set of parallel spaced-apart virtual scan lines oriented at a different angle from the previously processed set of virtual scan lines (e.g. 0, 30, 60, 90, 120 or 150 degrees). This processing cycle continues until a single bar code symbol is read within the processed image.

FIG. **21**B illustrates the steps involved in the image processing method carried out by the Multi-Mode Bar Code Symbol Subsystem **17** during its OmniScan Mode of operation. As indicated at Block A in FIG. **21**B, the Subsystem **17** first finds (i.e. calculates) the start pixel and scan angle in the captured narrow-area image. Then at Block B, the Subsystem **17** invokes the Decode Module and configures the same using the calculated (i) start pixel and (ii) start scan angle. Within the Decode Module, sub-Blocks B**1** through B**8** are then carried out as shown in FIG. **21**B. As indicated at Block B**1**, the Decoder Module, starting from the calculated start point and start angle, scans the image at the start angle and northwestwardly using a spot-size window of say N×N pixels (e.g. where 1<N<10), and then at Block B**2** processes the scanned image data to determine if a first border in a bar code symbol is found. Notably, this virtual scanning process is realized as a mathematical convolution of the spot-size window and the pixel data in the image buffer. If a first border is found at Block B**2**, then, once again starting from the calculated start point and start angle, the Decoder Module at Block B**3** scans the image at the start angle and southwestwardly using a spot size window of say N×N pixels (e.g. where 1<N<10), and then at Block B**4** processes the scanned image data to determine if a second border in a bar code symbol is found. If a second border is found at Block B**4**, then the Decoder Module invokes the Decoder Module described above at Block B**5** and decode processes the scanned image. If, at Block B**6**, the Decoder Module successfully reads a bar code symbol within the scanned line of image data, then the Subsystem **17** terminates the Decoder Module and stops the Omniscan Mode of operation.

If at Block B**2** in FIG. **21**A the Decoder Module does not find a first border of a bar code symbol, then it proceeds to Block B**7** and determines if it has tried all possible scan lines at combinations of start pixels and start angles within the captured narrow-area image. If at Block B**7** the Decoder Module has tried processing all possible scan lines at start pixel and angle combinations through the narrow-area image, then it proceeds to the "stop" Block and terminates the Omniscan Mode of decoder operation. If the Decoder Module has not tried processing all scan lines at all start pixel and angle orientations through the captured narrow-area image, then it proceeds to Block B**8**, where it advances to the next line of scan data in the captured narrow-area image (i.e. by the offset pixel amount n), and then returns to Block B**1** where scanning and processing is resumed along the new scan line (using a spot size window of say N×N pixels (e.g. where 1<N<10).

If at Block B**4**, the second bar code border is not found, then the Decoder Module proceeds to Block B**7** and determines whether all scan lines at all possible start pixels and angles (through the captured image) have been tried. If so, then the Decode Module terminates its process and exits the Omnsican Mode of operation. If the scan lines at all start pixel and angle combinations have not been tried at this stage of the process, then the Decoder Module proceeds to Block B**8** and advances the next start pixel and angle for scan data image processing, and returns to Block B**1** as described hereinabove.

If at Block G in FIG. **21**A the Decoder Module does not decode a bar code within the current set of parallel lines of scan data being processed, then it proceeds to Block I, where it advances to the next set of parallel scan lines (at a different set of start pixels and angle), and then returns to Block B where scanning and processing is resumed along the new set of parallel scan lines (using a spot size window of say N×N pixels (e.g. where 1<N<10). For every bar code that is decoded by the Decoder Module, a callback function is invoked to save the decoded result. The Decoder Module calls the Pause Checker callback function frequently to let the bar code reading Application take control.

Specification of Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its ROI-Specific Mode of Operation

FIG. **22**A illustrates the steps involved in the image processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its ROI-Specific Mode of operation. Notably, the ROI-Specific Mode of operation is similar to the Manual Mode of operation, except that it is used to automatically process a specified "region of interest" (ROI) previously identified during the processing of a captured image frame during a different mode of operation, e.g. the NoFinder Mode of Operation or Omniscan Mode of Operation

As reflected in FIG. **22**A, during this ROI-Specific Mode of operation, the first stage of processing involves receiving region of interest (ROI) coordinates {x,y} obtained during other modes of operation (e.g. Omniscan Mode, Automatic Mode or NoFinder Mode—after the occurrence of a failure to read), and re-partitioning the captured low-resolution image (from the Omniscan Mode) into N×N blocks, and instantiating a feature vector for the ROI-specified block(s) using features imported from and collected during the Omniscan, Automatic or No-Finder Module (and possibly utilizing additional spatial-derivative based image processing techniques). The second stage of processing involves marking additional ROIs by examining the feature vectors for regions of high-modulation (about the originally specified ROI) and returning to the first stage to create feature vectors for other blocks surrounding the specified block (in a helical manner), calculating bar code orientation and marking the four corners of a bar code contained within a ROI to be decode processed. The third stage of processing involves reading any bar code symbols represented within the ROI by traversing the pixel data associated with the bar code and updating the feature vectors, examining the zero-crossings of filtered images, creating bar and space patterns, and decoding the bar and space patterns using conventional bar code decoding algorithms.

FIG. **22**B illustrates the steps involved in the image processing method carried out by the Multi-Mode Bar Code Symbol Reading Subsystem during its ROI-Specific Mode of operation. As indicated at Block A, the Decoder Module associated with either the Omniscan or NoFinder Mode receives {x.y} coordinates for a specific ROI (in which at least a portion of a bar code symbol is likely to exist) to which an initial feature vector will be instantiated. Then at Block B, the Finder Mode is invoked, and at Block C, the Finder Module determines whether or not an ROI (containing a complete bar code symbol)) has been found. If the Finder Module determines that a ROI-contained bar code has been found, then the Finder Module invokes the Marker Module, whereupon at Block E, the Marker Module determines whether the ROI-contained bar code symbol has been marked by the Marker Module. If so, then the Decoder Module is invoked and then the high-resolution pixel data associated with the ROI is processed. If a bar code symbol is read within the ROI at Block G, then at Block H the Decoder Module determines if the actual number of decodes equals the required number of decode cycles (i.e. set by the end user). If so, then the Manual Mode of Operation is stopped, and the flow returns to the Application Layer.

If at Block C in FIG. **22**B the Finder Module determines that an ROI (containing a complete bar code) is not found,

then the Finder Module proceeds to Block I. If the Finder Mode determines that all feature vectors have not yet been examined, then the Finder Mode proceeds to Block J which advances the analysis to the next feature vector closet to the ROI-specified feature vector, along the locus of a helical path through the image pixel data set. Then, at Block B, the Finder Module reinvokes itself to operate on this next feature vector.

If at Block G, the Decoder Module does not successfully read a bar code symbol in the ROI, then it advances to Block I and determines whether all feature vectors have not been examined. If so, then the Decoder Module terminates the ROI-specific Mode of operation. Typically, the Subsystem **17** continues in this mode of operation until, for example, a single bar code symbol is read within an ROI marked as containing a complete bar code symbol. Each instance of the Finder Module involves the analysis of another block of pixel data (corresponding to another feature vector) in effort to find an ROI containing a complete bar code symbol, which can be found at Block C and successfully read at Block G. The sequential analysis of blocks of pixel data follows a helical pattern about the center starting point, determined at Block A of FIG. **22**B. Notably, during the Manual Mode of Operation, the Subsystem utilizes the image processing techniques described in connection with the Automatic Mode of Operation, above.

Specification of Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its First Multi-Read (Omniscan/ROI-Specific) Mode of Operation

FIG. **23** describes the operation of the Multi-Mode Bar Code Symbol Reading Subsystem **17** when it is driven into its first multi-read (e.g. Omniscan/ROI-Specific) mode of operation. In this first multi-read mode of operation, the Subsystem **17** adaptively processes and reads a captured high-resolution image in a high-speed manner, applying adaptive learning techniques, taught herein.

For example, assume the multi-mode image-processing symbol decoding subsystem is configured to operate in its first multi-read (OmniScan/ROI-Specific) mode of operation, as shown in FIG. **23**. In this case, if during the Omniscan Mode of operation, code fragments associated with a PDF417 bar code symbol are detected within a ROI in a captured (narrow or wide) area image, but processing thereof is unsuccessful, then the Multi-Mode Bar Code Symbol Reading Subsystem **17** will automatically (i) enter its ROI-Specific Mode of operation described above, and then (ii) immediately commences processing of the captured image at the ROI specified by ROI coordinates acquired by feature vector analysis during the Omniscan Mode of operation. In the illustrative embodiment, this switching of modes in the Subsystem **17** occurs within a single bar code symbol reading cycle, and involves processing a captured image frame using at least two different modes (i.e. methods) of image-processing based bar code reading, within which potentially dozens of different bar code symbol decoding algorithms are typically applied each decoding cycle.

One potential advantage of the Multi-Read (Omniscan/ROI-Specific) Mode of operation, over the Manual Mode of operation, is that the Multi-Read Mode offers an OmniScan Mode of operation to initially and rapidly read 1D bar code symbologies, and various kinds of 2D bar code symbologies whenever present in the captured image, and whenever a PDF417 symbology is detected (through its code fragments), the Multi-Mode Bar Code Symbol Reading Subsystem **17** can automatically switch (on-the-fly) to its ROI-specific Mode of operation to immediately process high-resolution image data at a specific ROI (at which there is a high likelihood of a bar code symbol present).

Specification of Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its Second Multi-Read (No-Finder/ROI-Specific) Mode of Operation

FIG. **24** illustrates the Multi-Mode Bar Code Symbol Reading Subsystem **17** when it is driven into its second multi-read (No-Finder/ROI-Specific) mode of operation so as to adaptively process and read a captured high-resolution image, in a high-speed manner, by applying adaptive learning techniques.

For example, assume the Multi-Mode Bar Code Symbol Reading Subsystem **17** is configured to operate in its second multi-read (No-Finder/ROI-Specific) mode when processing a wide-area image captured by the system, as shown in FIG. **24**. In this case, if during the NoFinder Mode of operation, code fragments associated with a PDF417 bar code symbol are detected within the captured wide-area image, but processing thereof is unsuccessful, then the Subsystem **17** will automatically (i) enter its ROI-specific mode of operation described above, and then (ii) immediately commence processing of the captured wide-area image at a ROI specified by y coordinates corresponding to the wide-area image processed during the NoFinder Mode of operation. In the illustrative embodiment, this switching of modes in the Image-Processing Bar Code Symbol Reading Subsystem **17** occurs within a single bar code symbol reading cycle, and involves processing a single captured image frame using at least two different modes (i.e. methods) of image-processing based bar code reading (i.e. NoFinder Mode and ROI-Specific), within each of which potentially dozens of different bar code symbol decoding algorithms are typically applied during each decoding cycle.

Alternatively, assume the Subsystem **17** is configured to operate in its "multi-read mode" when processing first a narrow-area image and then a wide-area image captured by the system. In this case, if during the NoFinder Mode of operation, code fragments associated with a PDF417 bar code symbol are detected within the captured narrow-area image, but decode processing thereof is unsuccessful, then the Subsystem **17** will automatically (i) enter its ROI-specific mode of operation described above, as a wide-area image is automatically captured by the system, and then (ii) immediately commence processing the captured wide-area image at a ROI specified by y coordinates corresponding to the narrow-area image processed during the NoFinder Mode of operation. In the illustrative embodiment, this switching of modes in the Subsystem **17** occurs within a single bar code symbol reading cycle, and involves processing two captured image frames using at least two different modes (i.e. methods) of image-processing based bar code reading (i.e. NoFinder Mode and ROI-Specific), within each of which potentially dozens of different bar code symbol decoding algorithms are typically applied during each decoding cycle.

One potential advantage of the "No-Finder/ROI-Specific" Multi-Mode operation over the Manual Mode of operation, regardless of its method of implementation, is that the No-Finder Mode can rapidly read 1D bar code symbologies whenever they are presented to the bar code symbol reader, and then whenever a 2D (e.g. PDF417) symbology is encountered, the bar code symbol reader can automatically switch its method of reading to the ROI-specific Mode use features collected from a narrow (or wide) area image processed during the No-Finder Mode, so as to immediately process a specific ROI in a captured wide-area image frame, at which

there is a high likelihood of a bar code symbol present, and to do so in a highly targeted manner.

Specification of Multi-Mode Bar Code Symbol Reading Subsystem of the Present Invention Operated During its Third Multi-Read (No-Finder/OmniScan/ROI-Specific) Mode of Operation

FIG. **25** illustrates the Multi-Mode Bar Code Symbol Reading Subsystem **17** when it is driven into its third multi-read (No-Finder/OmniScan/ROI-Specific) mode of operation so as to adaptively process and read a captured high-resolution image, in a high-speed manner, by applying adaptive learning techniques.

For example, assume the Subsystem **17** is configured to operate in its "multi-read mode" when processing a wide-area image captured by the system, as shown in FIG. **25**. In this case, if during the NoFinder Mode of operation, code fragments associated with a PDF417 bar code symbol are detected within the captured narrow-area image, but decode processing thereof is unsuccessful, then the Image Formation and Detection Subsystem (i) automatically captures a wide-area image, while the Subsystem **17** (ii) automatically enters its Omniscan Mode of operation described above, and then (iii) immediately commences processing of the captured wide-area image at a plurality of parallel spatially-separated (e.g. by 50 pixels) virtual scan lines, beginning at a start pixel and start angle specified by x and/or y coordinates of code fragments detected in the narrow-area image processed during the NoFinder Mode of operation. Then, if the Omniscan Mode does not successfully read a bar code symbol within the ROI, then the Subsystem **17** (ii) automatically enters its ROI-specific mode of operation described above, and then (iii) immediately commences processing of the captured wide-area image at a ROI specified by the x,y coordinates corresponding to code fragments detected in the wide-area image processed during the Omniscan Mode of operation. In the illustrative embodiment, this switching of modes in the Subsystem **17** occurs within a single bar code symbol reading cycle, and involves processing two captured image frames using at least three different modes (i.e. methods) of image-processing based bar code reading (i.e. NoFinder Mode, Omniscan Mode, and ROI-Specific Mode), within each of which potentially dozens of different bar code symbol decoding algorithms are typically applied during each decoding cycle.

One potential advantage of the "No-Finder/OmniScan/ROI-Specific" Multi-Read Mode operation over the Manual Mode of operation, regardless of its method of implementation, is that the No-Finder Mode can rapidly acquire 1D bar code symbologies whenever they are presented to the bar code symbol reader, and then whenever a 2D symbology is encountered, the bar code symbol reader can automatically switch its method of reading to the OmniScan Mode, collected features on processed image data, and if this decoding method is not successful, then the bar code reader can automatically switch its method of reading to the ROI-Specific Mode and use features collected during the Omniscan Mode to immediately process a specific ROI in a captured image frame, at which there is a high likelihood of a bar code symbol present, and to do so in a highly targeted manner.

Programmable Modes of Bar Code Reading Operation within the Hand-Supportable Digital Image-Based Bar Code Reading Device of the Present Invention

As indicated in FIG. **26**, the Imaging-Based Bar Code Symbol Reader of the present invention has at least seventeen (17) Programmable System Modes of Operation, namely: Programmed Mode of System Operation No. 1—Manually-Triggered Single-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode Of System Operation No. 2—Manually-Triggered Multiple-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode Of System Operation No. 3—Manually-Triggered Single-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode And The Automatic Or Manual Modes of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 4—Manually-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode And The Automatic Or Manual Modes of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 5—Manually-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing the No-Finder Mode And The Automatic Or Manual Modes of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 6—Automatically-Triggered Single-Attempt 1D Single-Read Mode Employing The No-Finder Mode Of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 7—Automatically-Triggered Multi-Attempt 1D Single-Read Mode Employing The No-Finder Mode Of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 8—Automatically-Triggered Multi-Attempt 1D/2D Single-Read Mode Employing The No-Finder Mode and Manual and/or Automatic Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 9—Automatically-Triggered Multi-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder Mode and Manual and/or Automatic Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of System Operation No. 10—Automatically-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing The Manual, Automatic or Omniscan Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmed Mode of System Operation No. 11—Semi-Automatic-Triggered Single-Attempt 1D/2D Single-Read Mode Employing The No-Finder Mode And The Automatic Or Manual Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of System Operation No. 12—Semi-Automatic-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing The No-Finder Mode And The Automatic Or Manual Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of Operation No. 13—Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder Mode And The Automatic Or Manual Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of Operation No. 14—Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The No-Finder Mode And The Omniscan Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of Operation No. 15—Continuously-Automatically-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing The Automatic, Manual Or Omniscan Modes Of the Multi-Mode Bar Code Reading Subsystem; Programmable Mode of System Operation No. 16—Diagnostic Mode Of Imaging-Based Bar Code Reader Operation; and Programmable Mode of System Operation No. 17—Live Video Mode Of Imaging-Based Bar Code Reader Operation.

Preferably, these Modes Of System Operation can programmed by reading a sequence of bar code symbols from a programming menu as taught, for example, in U.S. Pat. No. 6,565,005, which describes a bar code scanner programming technology developed by Metrologic Instruments, Inc., and

marketed under the name MetroSelect® Single Line Configuration Programming Method.

These Programmable System Modes of Operation will be described in detail herein below. Alternatively, the MetroSet® Graphical User Interface (GUI) can be used to view and change configuration parameters in the bar code symbol reader using a PC. Alternatively, a Command Line Interface (CLI) may also be used to view and change configuration parameters in the bar code symbol reader,

Each of these programmable modes of bar code reader operation shall be now described in greater detail with reference to other components of the system that are configured together to implement the same in accordance with the principles of the present invention.

Overview of the Imaging-Based Bar Code Reader Start-Up Operations

When the bar code reader hereof boots up, its FPGA is programmed automatically with 12.5/50/25 MHz clock firmware and all required device drivers are also installed automatically. The login to the Operating System is also done automatically for the user "root", and the user is automatically directed to the /root/ directory. For nearly all programmable modes of system operation employing automatic object detection, the IR object detection software driver is installed automatically. Also, for all Programmable System Modes of operation employing the narrow-area illumination mode, the narrow-area illumination software drivers are automatically installed, so that a Pulse Width Modulator (PWM) is used to drive the narrow-area LED-based illumination array **27**. To start the bar code reader operation, the operating system calls the /tmp/ directory first ("cd /tmp"), and then the focusapp program, located in /root/ directory is run, because the /root/ directory is located in Flash ROM, and to save captured images, the directory /tmp/ should be the current directory where the image is stored in transition to the host), which is located in RAM.

Programmed Mode of System Operation No. 1: Manually-Triggered Single-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 1 involves configuration of the system as follows: disabling the IR-based Object Presence and Range Detection Subsystem **12**; and enabling the use of manual-trigger activation, the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**.

During this mode of system operation, when a user pulls the trigger switch **2**C, the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem **17**. Then, the bar code reader illuminates the target object using narrow-area illumination, captures a narrow-area image of the target object, and launches the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem **17**. The captured image is then processed using the No-Finder Mode. If a single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If single cycle of programmed image processing is not result in a successful reading of a 1D bar code symbol, then the cycle is terminated, all subsystems are deactivated, and the bar code reader returns to its sleep mode of operation, and wait for the next event (e.g. manually pulling trigger switch **2**C) which will trigger the system into active operation.

Programmed Mode Of System Operation No. 2: Manually-Triggered Multiple-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 2 involves configuration of the system as follows: disabling the IR-based Object Presence and Range Detection Subsystem **12**; and enabling the use of manual-trigger activation, the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem **17**.

During this mode of system operation, when a user pulls the trigger switch **2**C, the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, the bar code reader illuminates the target object using narrow-area illumination, captures a narrow-area image of the target object, and launches the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem **17**. The captured narrow-area image is then processed using the No-Finder Mode. If the single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem for use by the host system. If the cycle of programmed image processing does not produce a successful read, then the system automatically enables successive cycles of illumination/capture/processing so long as the trigger switch **2**C is being pulled, and then until the system reads a bar code symbol within a captured image of the target object; only thereafter, or when the user releases the trigger switch **2**C, will the bar code symbol reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the trigger switch **2**C is being pulled by the user, the Imaging-Based Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the trigger switch **2**C is manually released.

Programmed Mode of System Operation No. 3: Manually-Triggered Single-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode and the Automatic, Manual or ROI-Specific Modes of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 3 involves configuration of the system as follows: disabling the IR-based Object Presence and Range Detection Subsystem **12**; and enabling the use of manual-trigger activation, the narrow-area and wide-area illumination modes within the Multi-Mode Illumination Subsystem **14**, the narrow-area and wide-area image capture modes in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmable mode of system operation, the bar code reader is idle (in its sleep mode) until a user points the bar code reader towards an object with a bar code label, and then pulls the trigger switch **2**C. When this event occurs, the system activates the narrow-area illumination mode

within the Multi-Mode Illumination Subsystem **14** (i.e. drives the narrow-area illumination array **27**), the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, the bar code reader illuminates the target object using narrow-area illumination, captures a narrow-area image of the target object, and launches the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured narrow-area image is then processed using the No-Finder Mode. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode of the Image Formation and Detection Subsystem **13**, and the Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**. Then the bar code reader illuminates the target object using both near-field and far-field wide-area illumination, captures a wide-area image of the target object, and launches the Manual, ROI-Specific or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured wide-area image is then processed using the Manual, ROI-Specific or Automatic Mode. If this single cycle of programmed image processing results in the successful reading of a 1D or 2D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the subsystem **19** deactivates all subsystems and then returns to its sleep mode, and waits for an event, which will cause it to re-enter its active mode of operation.

Programmed Mode of System Operation No. 4: Manually-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode and the Automatic, Manual or ROI-Specific Modes of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 4 involves configuration of the system as follows: disabling the IR-based object detection subsystem **12**; and enabling the use of manual-trigger activation, the narrow-area and wide-area illumination modes within the Multi-Mode Illumination Subsystem **14**, the narrow-area and wide-area image capture modes of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, when a user pulls the trigger switch **2**C, the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, the bar code reader illuminates the target object using narrow-area illumination, captures a narrow-area image of the target object, and launches the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured narrow-area image is then processed using the No-Finder Mode. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode of the Image Formation and Detection Subsystem **13**, and the Manual and/or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, the bar code reader illuminates the target object using both near-field and far-field wide-area illumination, captures a wide-area image of the target object, and launches the Manual (or Automatic) Mode of the Multi-Mode Bar Code Reading Subsystem. The captured wide-area image is then processed using the Manual Mode of bar code symbol reading. If this single cycle of programmed processing results in the successful reading of a 1D or 2D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read of a single 1D or 2D bar code symbol, then the Subsystem **19** automatically enables successive cycles of wide-area illumination/wide-area image capture and processing so long as the trigger switch **2**C is being pulled, and then until the system reads a single 1D or 2D bar code symbol within a captured image of the target object; only thereafter, or when the user releases the trigger switch **2**C, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the trigger switch is being pulled by the user, the Imaging-Based Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the trigger switch **2**C is manually released.

Programmed Mode of System Operation No. 5: Manually-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing the No-Finder Mode and the Automatic Manual or ROI-Specific Modes of the Multi-Mode Bar Code Reading Symbol Subsystem

Programmed Mode of System Operation No. 5 involves configuration of the system as follows: disabling the IR-based Object Presence and Range Detection Subsystem **12**; and enabling the use of manual-trigger activation, the narrow-area and wide-area illumination modes within the Multi-Mode Illumination Subsystem **14**, the narrow-area and wide-area image capture modes of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this mode of system operation, when a user pulls the trigger switch **2**C, the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem. Then, the bar code reader illuminates the target object using narrow-area illumination, captures a narrow-area image of the target object, and launches the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem. The captured

narrow-area image is then processed using the No-Finder Mode. If this single cycle of programmed processing results in the successful decoding of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed decode image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem, the wide-area image capture mode of the Image Formation and Detection Subsystem **13**, and the Manual and/or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, the bar code reader illuminates the target object using both near-field and far-field wide-area illumination, captures a wide-area image of the target object, and launches the Manual (ROI-Specific and/or Automatic) Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured wide-area image is then processed using the Manual Mode of reading. If this single cycle of programmed processing results in the successful reading of a 1D or 2D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful reading of one or more 1D and/or 2D bar code symbols, then the system automatically enables successive cycles of wide-area illumination/wide-area image capture/image processing so long as the trigger switch is being pulled, and then until the system reads one or more 1D and/or 2D bar code symbols within a captured image of the target object; only thereafter, or when the user releases the trigger switch **2**C, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the trigger switch **2**C is being pulled by the user, the Imaging-Based Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the trigger switch **2**C is manually released. Programmed Mode of System Operation No. 6: Automatically-Triggered Single-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 6 involves configuration of the system as follows: disabling the use of manual-trigger activation; and enabling IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area illumination mode only within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode only in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the reader towards an object with a bar code label. Once the object is under the field-of-view of the bar code reader, and the object is automatically detected, the bar code reader "wakes up" and the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, the system captures/acquires a narrow-area image, which is then processed using the Bar Code Symbol Reading Subsystem **17** configured in its No-Finder Mode of operation. If this single cycle of programmed decode processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates all subsystems, causing the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation.

Programmed Mode of System Operation No. 7: Automatically-Triggered Multi-Attempt 1D Single-Read Mode Employing the No-Finder Mode of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 7 involves configuration of the system as follows: disabling the use of manual-trigger activation; and enabling IR-based Object Presence And Range Detection Subsystem **12**, the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the bar code reader towards an object with a bar code label. Once the object is under the field-of-view of the bar code reader, and the object is automatically detected, the bar code reader "wakes up" and the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, the system captures/acquires a narrow-area image, which is then processed using the No-Finder Mode. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful decode, then the system automatically enables successive cycles of narrow-area illumination/narrow-area image capture/processing so long as the trigger switch **2**C is being pulled, and then until the system reads a single 1D bar code symbol within a captured image of the target object; only thereafter, or when the user releases the trigger switch **2**C, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the trigger switch is being pulled by the user, the Imaging-Based Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the trigger switch **2**C is manually released.

Programmed Mode of System Operation No. 8: Automatically-Triggered Multi-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 8 involves configuration of the system as follows: disabling the use of manual-trigger activation during all phase of system operation; and enabling IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the reader towards an object with a bar code label. Once the object is under the field-of-view of the scanner, and the object is automatically detected, the bar code reader "wakes up" and the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem, **14** the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, the system captures/acquires a narrow-area image, which is then processed using the No-Finder Mode of operation. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode in the Image Formation and Detection Subsystem **13**, and the Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**. Then, the Bar Code Symbol Reader illuminates the target object using either near-field or far-field wide-area illumination (depending on the detected range of the target object), captures a wide-area image of the target object, and launches the Manual, ROI-Specific or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured wide-area image is then processed using the Manual Mode of reading. If this cycle of programmed image processing results in the successful reading of a single 1D or 2D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful reading of a single 1D or 2D bar code symbol, then the system automatically enables successive cycles of wide-area illumination/wide-area image capture/processing so long as the target object is being detected, and then until the system reads one or more 1D and/or 2D bar code symbols within a captured image of the target object; only thereafter, or when the user moves the object out of the FOV of the bar code reader, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the object is being detected by the bar code reader, the Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the object is moved away from the FOV of the bar code reader.

Programmed Mode of System Operation No. 9: Automatically-Triggered Multi-Attempt 1D/2D Multiple-Read Mode Employing the No-Finder Mode and Manual ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 9 involves configuration of the system as follows: disabling the use of manual-trigger activation during all phases of system operation; and enabling IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the No Finder Mode and Manual or Automatic Modes of the Multi-Mode Bar Code Symbol Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the reader towards an object with a bar code label. Once the object is under the field-of-view of the bar code reader, and the object is automatically detected, the bar code reader "wakes up" and the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, the system captures/acquires a narrow-area image, which is then processed using the No-Finder Mode. If this single cycle of programmed processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode in the Image Formation and Detection Subsystem **13**, and the Manual and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**. Then, the bar code reader illuminates the target object using either near-field or far-field wide-area illumination (depending on the detected range of the target object), captures a wide-area image of the target object, and launches the Manual (ROI-Specific or Automatic) Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured wide-area image is then processed using the Manual Method of decoding. If this cycle of programmed image processing results in the successful reading of a single 1D or 2D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image process-

ing does not produce a successful read of a single 1D or 2D bar code symbol, then the system automatically enables successive cycles of wide-area-illumination/wide-area image-capture/processing so long as the target object is being detected, and then until the system reads one or more 1D and/or 2D bar code symbols within a captured image of the target object; only thereafter, or when the user moves the object out of the FOV of the bar code symbol reader, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the object is being detected by the bar code reader, the bar code reader will re-attempt reading every 500 ms (at most) until it either succeeds or the object is moved away from the FOV of the bar code reader.

Programmable Mode of System Operation No. 10: Automatically-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing the Manual, ROI-Specific, Automatic or Omniscan Modes of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 10 involves configuration of the system as follows: disabling the use of manual-trigger activation during all phase of system operation; and enabling IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the Manual, ROI-Specific, Automatic or OmniScan Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user present an object with a bar code symbol under the field-of-view of the bar code reader, and the object is automatically detected, the bar code reader "wakes up" and the system activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode in the Image Formation and Detection Subsystem **13**, and either Manual, ROI-Specific, Automatic or Omniscan Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a wide area of the target object within the field-of-view (FOV) of the bar code reader with far-field or near-field wide area illumination (depending on the detected range of the target object), and capture/acquire a wide-area image which is then processed using either the Manual, ROI-Specific, Automatic or Omniscan Method of reading. If this single cycle of programmed processing results in the successful reading of a 1D or 2D bar code symbol (when the Manual, ROI-Specific and Automatic Methods are used), then the resulting symbol character data is sent to the Input/Output Subsystem for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system automatically enables successive cycles of wide-area illumination/wide-area-image-capture/processing so long as the target object is being detected, and then until the system reads a single 1D and/or 2D bar code symbol within a captured image of the target object; only thereafter, or when the user moves the object out of the FOV of the bar code reader, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the object is being detected by the bar code reader, the bar code reader will re-attempt reading every 500 ms (at most) until it either succeeds or the object is moved away from the FOV of the bar code reader.

Programmed Mode of System Operation No. 11: Semi-Automatic-Triggered Single-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode and the Automatic, ROI-Specific or Manual Modes of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 11 involves configuration of the system as follows: disabling the use of the manual-trigger activation during the system activation phase of operation; and enabling the IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area and wide-area illumination modes within the Multi-Mode Illumination Subsystem **14**, the narrow-area and wide-area image capture modes in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the reader towards an object with a bar code label. Once the object is under the field-of-view of the bar code reader, and the object is automatically detected, the bar code reader "wakes up" and the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, the system captures/acquires a narrow-area image, which is then processed using the No-Finder Mode. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode of the Image Formation and Detection Subsystem **13**, and the Manual, ROI-Specific and/or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, if the user pulls the trigger switch **2C** during narrow-area illumination and image capture and continues to do so, the bar code reader will automatically illuminate the target object using wide-area illumination, capture a wide-area image of the target object, and launch the Manual, ROI-Specific or Automatic Mode of the Multi-Mode Bar Code Symbol Reading Subsystem **17**. The captured wide-area image is then processed using the Manual, ROI-Specific or Automatic Mode/Method of bar code reading. If this single cycle of programmed image processing results in the successful reading of a single 1D or 2D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful reading of a single 1D or 2D bar code symbol, then the subsystem **19** automatically deactivates all

subsystems, causing the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation

Programmable Mode of System Operation No. 12: Semi-Automatic-Triggered Multiple-Attempt 1D/2D Single-Read Mode Employing the No-Finder Mode and the Automatic, ROI-Specific or Manual Modes of the Multi-Mode Bar Code Symbol Reading Subsystem;

Programmed Mode of System Operation No. 12 involves configuration of the system as follows: disabling the use of manual-trigger activation during the system activation phase of operation; and enabling the IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area and wide-area illumination modes within the Multi-Mode Illumination Subsystem **14**, the narrow-area and wide-area image capture modes in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the reader towards an object with a bar code label. Once the object is under the field-of-view of the bar code reader, and the object is automatically detected, the bar code reader "wakes up" and the system activates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, the system captures/acquires a narrow-area image, which is then processed using the No-Finder Mode. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem, the wide-area image capture mode of the Image Formation and Detection Subsystem **13**, and the Manual, ROI-Specific and/or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, if the user pulls the trigger switch **2C** during narrow-area illumination and image capture and continues to do so, the bar code reader will automatically illuminate the target object using wide-area illumination, capture a wide-area image of the target object, and launches the Manual, ROI-Specific or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured wide-area image is then processed using the Manual Mode of reading. If this single cycle of programmed image processing results in the successful reading of a single 1D or 2D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful decode of a single 1D or 2D bar code symbol, then the system automatically enables successive cycles of wide-area illumination/wide-area-image-capture/processing so long as the trigger switch **2C** is being pulled, and then until the system reads one or more 1D and/or 2D bar code symbols within a captured image of the target object; only thereafter, or when the user releases the trigger switch **2C**, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the trigger switch **2C** is being pulled by the user, the Imaging-Based Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the trigger switch **2C** is manually released.

Implementation of Programmable Mode of System Operation No. 12

When the Focus IR module detects an object in front of object detection field **20**, it posts the OBJECT_DETECT_ON event to the Application Layer. The Application Layer software responsible for processing this event starts the CodeGate Task. When the user pulls the trigger switch **2C**, the TRIGGER_ON event is posted to the Application. The Application Layer software responsible for processing this event checks if the CodeGate Task is running and if so, it cancels it and then starts the Main Task. When the user releases the trigger switch **2C**, the TRIGGER_OFF event is posted to the Application. The Application Layer software responsible for processing this event, checks if the Main Task is running, and if so, it cancels it. If the object is still within the object detection field **20**, the Application Layer starts the CodeGate Task again.

When the user moves the bar code reader away from the object (or the object away from the bar code reader), the OBJECT_DETECT_OFF event is posted to the Application Layer. The Application Layer software responsible for processing this event checks if the CodeGate Task is running, and if so, it cancels it. The CodeGate Task, in an infinite loop, does the following. It activates the narrow-area illumination array **27** which illuminates a "narrow" horizontal area at the center of the field-of-view and then the Image Formation and Detection Subsystem **13** acquires an image of that narrow-area (i.e. few rows of pixels on the CMOS image sensing array **22**), and then attempts to read a bar code symbol represented in the image. If the read is successful, it saves the decoded data in the special CodeGate data buffer. Otherwise, it clears the CodeGate data buffer. Then, it continues the loop. The CodeGate Task never exits on its own; it can be canceled by other modules of the Focus software when reacting to other events.

When a user pulls the trigger switch **2C**, the event TRIGGER_ON is posted to the Application Layer. The Application Layer software responsible for processing this event, checks if the CodeGate Task is running, and if so, it cancels it and then starts the Main Task. The CodeGate Task can also be canceled upon OBJECT_DETECT_OFF event, posted when the user moves the bar code reader away from the object, or the object away from the bar code reader.

Programmable Mode of Operation No. 13: Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing the No-Finder Mode and the Automatic, ROI-Specific or Manual Modes of the Multi-Mode Bar Code Reading Subsystem

Programmed Mode of System Operation No. 13 involves configuration of the system as follows: disabling the use of manual-trigger activation during the system activation phase of operation; and enabling the IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area and wide-area illumination modes within the Multi-Mode Illumination

Subsystem **14**, the narrow-area and wide-area image capture modes in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and Manual, ROI-Specific and/or Automatic Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the reader towards an object with a bar code label. Once the object is under the field-of-view of the bar code reader, and the object is automatically detected by the Object Presence and Range Detection Subsystem **12**, the bar code reader "wakes up" and the system activates the narrow-area illumination mode in the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the system to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, the system captures/acquires a narrow-area image which is then processed using the No-Finder Mode. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful read, then the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**, and then activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode of the Image Formation and Detection Subsystem **13**, and the Manual and/or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. Then, if the user pulls the trigger switch **2**C during narrow-area illumination and image capture and continues to do so, the bar code reader will automatically illuminate the target object using wide-area illumination, capture a wide-area image of the target object, and invoke the Manual, ROI-Specific and/or Automatic Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured wide-area image is then processed using the Manual, ROI-Specific or Automatic Mode of reading. If this single cycle of programmed image processing results in the successful reading of one or more 1D and/or 2D bar code symbols, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed decode image processing does not produce a successful reading of one or more 1D and/or 2D bar code symbols then the system automatically enables successive cycles of wide-area illumination/wide-area-image-capture/image-processing so long as the trigger switch **2**C is being pulled, and then until the system reads one or more 1D and/or 2D bar code symbols within a captured image of the target object; only thereafter, or when the user releases the trigger switch **2**C, will the bar code reader return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the trigger switch **2**C is being pulled by the user, the Imaging-Based Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the trigger switch **2**C is manually released.

Programmable Mode of Operation No. 14: Semi-Automatic-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing the No-Finder Mode and the Omniscan Modes of the Multi-Mode Bar Code Symbol Reading Subsystem

Programmed Mode of System Operation No. 14 involves configuration of the system as follows: disabling the use of manual-trigger activation during the system activation phase of operation; and enabling the IR-based Object Presence and Range Detection Subsystem **12**, the narrow-area and wide-area illumination modes within the Multi-Mode Illumination Subsystem **14**, the narrow-area and wide-area image capture modes in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode and OmniScan Mode of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader is idle until a user points the reader towards an object with a bar code label. Once the object is under the field-of-view of the bar code reader, and the object is automatically detected by the Object Presence and Range Detection Subsystem **12**, the bar code reader "wakes up" and the system activates the narrow-area illumination mode in the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode in the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**. This causes the narrow-area illumination array **27** to illuminate a "narrow" horizontal area of the target object at the center of the field-of-view (FOV) of the bar code reader, indicating to the user where the area targeted by the bar code reader is, and thus, enabling the user to position and align the narrow-area illumination beam on the target bar code. Then, Subsystem **13** captures/acquires a narrow-area image which is then processed by Subsystem **17** using its No-Finder Mode. If this single cycle of programmed image processing results in the successful reading of a 1D bar code symbol, then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system, and then the system deactivates all subsystems and resumes its sleep state of operation. If this cycle of programmed image processing does not produce a successful read, it may nevertheless produce one or more code fragments indicative of the symbology represented in the image, (e.g. PDF417). In this case, the system deactivates the narrow-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the narrow-area image capture mode of the Image Formation and Detection Subsystem **13**, and the No-Finder Mode of the Multi-Mode Bar Code Reading Subsystem **17**; and then, if the user is pulling the trigger switch **2**C at about this time, the system activates the wide-area illumination mode within the Multi-Mode Illumination Subsystem **14**, the wide-area image capture mode of the Image Formation and Detection Subsystem, and either the Omniscan Mode of the Multi-Mode Bar Code Reading Subsystem **17** if code fragments have been found indicating a 2D code format (e.g. PDF format code) within the image at perhaps a particular orientation. Then, the bar code reader proceeds to automatically illuminate the target object using wide-area illumination, capture a wide-area image of the target object, and invoke the Omniscan Mode of the Multi-Mode Bar Code Reading Subsystem **17**. The captured wide-area image is then first processed using the Omniscan Mode, using a first processing direction (e.g. at 0 degrees), and sequentially advances the Omniscan Mode of reading at an different angular orientation (e.g. 6 possible directions/orientations) until a single bar code symbol is successfully read. If this single cycle of programmed decode processing (using the Omniscan Mode) results in the successful decoding of a single 1D and/or 2D bar code symbol, then the resulting symbol character data is sent

to the Input/Output Subsystem **18** for use by the host system. If this cycle of programmed image processing does not produce a successful reading of a single 1D and/or 2D bar code symbol, then the system automatically enables successive cycles of wide-area illumination/wide-area image capture/processing so long as the trigger switch **2**C is being pulled, and then until the system reads a single 1D and/or 2D bar code symbol within a captured image of the target object. Only thereafter, or when the user releases the trigger switch **2**C, the system will return to its sleep mode of operation, and wait for the next event that will trigger the system into active operation. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the trigger switch **2**C is being pulled by the user, the Imaging-Based Bar Code Symbol Reader will re-attempt reading every 500 ms (at most) until it either succeeds or the trigger switch is manually released.

Programmable Mode of Operation No. 15: Continuously-Automatically-Triggered Multiple-Attempt 1D/2D Multiple-Read Mode Employing the Automatic, Manual ROI-Specific or Omniscan Modes of the Multi-Mode Bar Code Reading Subsystem

Programmed Mode of System Operation No. 15, typically used for testing purposes, involves configuration of the system as follows: disabling the use of manual-trigger activation during all phase of system operation; and enabling IR-based Object Presence and Range Detection Subsystem **12**, the wide-area illumination mode in the Multi-Mode Illumination Subsystem, **14** the wide-area image capture mode in the Image Formation and Detection Subsystem **13**, and the Manual, ROI-Specific, Automatic or OmniScan Modes of the Multi-Mode Bar Code Reading Subsystem **17**.

During this programmed mode of system operation, the bar code reader continuously and sequentially illuminates a wide area of the target object within the field-of-view (FOV) of the bar code reader with both far-field and near-field wide-area illumination, captures a wide-area image thereof, and then processes the same using either the Manual, ROI-Specific, Automatic or Omniscan Modes of operation. If any cycle of programmed image processing results in the successful reading of a 1D or 2D bar code symbol (when the Manual, ROI-Specific and Automatic Modes are used), then the resulting symbol character data is sent to the Input/Output Subsystem **18** for use by the host system (i.e. typically a test measurement system). If when any cycle of programmed image processing does not produce a successful read, the system automatically enables successive cycles of wide-area illumination/wide-area image-capture/processing. In the illustrative embodiment, the default decode timeout is set to 500 ms which can be simply changed by programming. This default decode timeout setting ensures that while the object is being detected by the bar code reader, the bar code reader will re-attempt reading every 500 ms (at most) until it either succeeds or the object is moved away from the FOV of the bar code reader.

Diagnostic Mode of Imaging-Based Bar Code Reader Operation: Programmable Mode of System Operation No. 16

Programmed Mode of System Operation No. 16 is a Diagnostic Mode. An authorized user can send a special command to the bar code reader to launch a Command Line Interface (CLI) with the bar code reader. When the bar code reader receives such request from the user, it sends a prompt "MTLG>" back to the user as a handshaking indication that the scanner is ready to accept the user commands. The user then can enter any valid command to the bar code reader and view the results of its execution. To communicate with the reader in diagnostic mode over such communication line as RS232, the user can use any standard communication program, such as Windows HyperTerminal for example. This mode of operation can be used to test/debug the newly introduced features or view/change the bar code reader configuration parameters. It can also be used to download images and/or a backlog of the previously decoded bar code data from the reader memory to the host computer.

Live Video Mode of Imaging-Based Bar Code Reader Operation: Programmable Mode of System Operation No. 17

Program Mode of System Operation No. 17 can be used in combination with any other supported imaging modes. In this mode, the images acquired by the bar code reader are transmitted to the host computer in real-time along with the results of image-processing based bar code symbol reading by Subsystem **17** (if such results are available).

Second Illustrative Embodiment of Digital Imaging-Based Bar Code Symbol Reading Device of the Present Invention, Wherein Four Distinct Modes of Illumination are Provided

In the first illustrative embodiment described above, the Multi-mode Illumination Subsystem **14** had three primary modes of illumination: (1) narrow-area illumination mode; (2) near-field wide-area illumination mode; and (3) far-field wide-area illumination mode.

In a second alternative embodiment of the Digital Imaging-Based Bar Code Symbol Reading Device of the present invention shown in FIGS. **27**A, **27**B and **28**, the Multi-Mode Illumination Subsystem **14** is modified to support four primary modes of illumination: (1) near-field narrow-area illumination mode; (2) far-field narrow-area illumination mode; (3) near-field wide-area illumination mode; and (4) far-field wide-area illumination mode. In general, these near-field and far-field narrow-area illumination modes of operation are conducted during the narrow-area image capture mode of the Multi-Mode Image Formation and Detection Subsystem **13**, and are supported by a near-field narrow-illumination array **27**A and a far field narrow-area illumination array **27**B illustrated in FIG. **28**, and as shown in FIG. **2**A**1**. In the second illustrative embodiment, each of these illumination arrays **27**A, **27**B are realized using at least a pair of LEDs, each having a cylindrical lens of appropriate focal length to focus the resulting narrow-area (i.e. linear) illumination beam into the near-field portion **24**A and far-field portion **24**B of the field of view of the system, respectively.

One of advantages of using a pair of independent illumination arrays to produce narrow-area illumination fields over near and far field portions of the FOV is that it is possible to more tightly control the production of a relatively "narrow" or "narrowly-tapered" narrow-area illumination field along its widthwise dimension. For example, as shown in FIG. **27**B, during bar code menu reading applications, the near-field narrow area illumination array **27**A can be used to generate (over the near-field portion of the FOV) an illumination field **24**A that is narrow along both its widthwise and height-wise dimensions, to enable the user to easily align the illumination field (beam) with a single bar code symbol to be read from a bar code menu of one type or another, thereby avoiding inadvertent reads of two or more bar code symbols or simply the wrong bar code symbol. At the same time, the far-field narrow area illumination array **27**B can be used to generate (over the far-field portion of the FOV) an illumination field **24**B that is sufficient wide along its widthwise dimension, to enable the user to easily read elongated bar code symbols in the far-field portion of the field of view of the bar code reader, by simply moving the object towards the far portion of the field.

Third Illustrative Embodiment of Digital Imaging-Based Bar Code Symbol Reading Device of the Present Invention

Alternatively, the Imaging-Based Bar Code Symbol Reading Device of the present invention can have virtually any type of form factor that would support the reading of bar code symbols at diverse application environments. One alternative form factor for the bar code symbol reading device of the present invention is shown in FIGS. **29**A through **29**C, wherein a portable Digital Imaging-Based Bar Code Reading Device of the present invention **1"** is shown from various perspective views, while arranged in a Presentation Mode (i.e. configured in Programmed System Mode No. 12).

The Digital Imaging-Based Bar Code Reading Device of the Present Invention

As shown in FIG. **30**, the Digital Imaging-Based Bar Code Reading Device of the present invention **1'**, **1"** can also be realized in the form of a Digital Imaging-Based Bar Code Reading Engine **100** that can be readily integrated into various kinds of information collection and processing systems. Notably, trigger switch **2**C shown in FIG. **30** is symbolically represented on the housing of the engine design, and it is understood that this trigger switch **2**C or functionally equivalent device will be typically integrated with the housing of the resultant system into which the engine is embedded so that the user can interact with and actuate the same. Such Engines according to the present invention can be realized in various shapes and sizes and be embedded within various kinds of systems and devices requiring diverse image capture and processing functions as taught herein.

Illustrative Embodiment of a Wireless Bar Code-Driven Portable Data Terminal (PDT) System of the Present Invention

FIGS. **31**, **32** and **33** show a Wireless Bar Code-Driven Portable Data Terminal (PDT) System **140** according to the present invention which comprises: a Bar Code Driven PDT **150** embodying the Digital Imaging-Based Bar Code Symbol Reading Engine of the present invention **100**, described herein; and a cradle-providing Base Station **155**.

As shown in FIGS. **31** and **32**, the Digital Imaging-Based Bar Code Symbol Reading Engine **100** can be used to read bar code symbols on packages and the symbol character data representative of the read bar code can be automatically transmitted to the cradle-providing Base Station **155** by way of an RF-enabled 2-way data communication link **170**. At the same time, robust data entry and display capabilities are provided on the PDT **150** to support various information based transactions that can be carried out using System **140** in diverse retail, industrial, educational and other environments.

As shown in FIG. **32**, the Wireless Bar Code Driven Portable Data Terminal System **140** comprises: a hand-supportable housing **151**; Digital Imaging-Based Bar Code Symbol Reading Engine **100** as shown in FIG. **30**, and described herein above, mounted within the head portion of the hand-supportable housing **151**; a user control console **151**A; a high-resolution color LCD display panel **152** and drivers mounted below the user control console **151**A and integrated with the hand-supportable housing, for displaying, in a real-time manner, captured images, data being entered into the system, and graphical user interfaces (GUIs) generated by the end-user application running on the virtual machine of the wireless PDT; and PDT computing subsystem **180** contained within the PDT housing, for carrying out system control operations according to the requirements of the end-user application to be implemented upon the hardware and software platforms of the wireless PDT **2**B of this illustrative embodiment.

As shown in block schematic diagram of FIG. **34**, a design model for the Wireless Hand-Supportable Bar Code Driven Portable Data Terminal System **140** shown in FIGS. **31** and **32**, and its cradle-supporting Base Station **155** interfaced with possible host systems **173** and/or networks **174**, comprises a number of subsystems integrated about a system bus, namely: a data transmission circuit **156** for realizing the PDT side of the electromagnetic-based wireless 2-way data communication link **170**; program memory (e.g. DRAM) **158**; non-volatile memory (e.g. SRAM) **159**; Digital Imaging-Based Bar Code Symbol Reading Engine **100** for optically capturing narrow and wide area images and reading bar code symbols recognized therein; a manual data entry device such as a membrane-switching type keypad **160**; LCD panel **152**; an LCD controller **161**; LCD backlight brightness control circuit **162**; and a system processor **163** integrated with a systems bus (e.g. data, address and control buses). Also, a battery power supply circuit **164** is provided for supplying regulated power supplies to the various subsystems, at particular voltages determined by the technology used to implement the PDT device.

As shown in FIG. **34**, the Base Station **155** also comprises a number of integrated subsystems, namely: a data receiver circuit **165** for realizing the base side of the electromagnetic-based wireless 2-way data communication link **170**; a data transmission subsystem **171** including a communication control module; a base station controller **172** (e.g. programmed microcontroller) for controlling the operations of the Base Station **155**. As shown, the data transmission subsystem **171** interfaces with the host system **173** or network **174** by way of the USB or RS232 communication interfaces, TCP/IP, AppleTalk or the like, well known in the art. Taken together, data transmission and reception circuits **156** and **165** realize the wireless electromagnetic 2-way digital data communication link **170** employed by the wireless PDT of the present invention.

Notably, Wireless Hand-Supportable Bar Code Driven Portable Data Terminal System **140**, as well as the POS Digital Imaging-Based Bar Code Symbol Reader **1"** shown in FIGS. **29**A through **29**C, each have two primary modes of operation: (1) a hands-on mode of operation, in which the PDT **150** or POS Reader **1"** is removed from its cradle and used as a bar code driven transaction terminal or simply bar code symbol reader; and (2) a hands-free mode of operation, in which the PDT **150** or POS Reader **1"** remains in its cradle-providing Base Station **155**, and is used a presentation type bar code symbol reader, as required in most retail point-of-sale (POS) environments. Such hands-on and hands-free modes of system operation are described in greater detail in copending U.S. patent application Ser. No. 10/684,273 filed on Oct. 11, 2003, and incorporated herein by reference in its entirety.

In such hands-on and hands-free kinds of applications, the trigger switch **2**C employed in the Digital Imaging Bar Code Symbol Reading Device of the present invention can be readily modified, and augmented with a suitable stand-detection mechanism, which is designed to automatically configure and invoke the PDT **150** and its Engine **100** into its Presentation Mode (i.e. System Mode of Operation No. 12) or other suitable system mode when the PDT is placed in its Base Station **155** as shown in FIG. **33**. Then when the PDT **150** is picked up and removed from its cradling supporting Base Station **155** as shown in FIGS. **31** and **32**, the trigger switch **2**C and stand-detection mechanism, arrangement can be arranged so as to automatically configure and invoke the PDT **150** and its Engine **100** into a suitable hands-on support-

ing mode of system operation (selected from the Table set forth in FIGS. **26**A and **26**B), to enable hands-on mode of operation.

Similarly, the trigger switch **2**C employed in the POS Digital Imaging Bar Code Symbol Reading Device **1"** can be readily modified, and augmented with stand-detection mechanism, which is designed to automatically configure and invoke the POS Reader **1"** into its Presentation Mode (i.e. System Mode of Operation No. 12) or other suitable system mode, when the Reader **1"** is resting on a countertop surface, as shown in FIGS. **29**A and **29**B. Then when the POS Reader **1"** is picked up off the countertop surface, for use in its hands-on mode of operation, the trigger switch **2**C and stand-detection mechanism, arrangement will automatically configure and invoke Reader **1"** into a suitable hands-on supporting mode of system operation, as shown in FIG. **29**C. In such embodiments, the stand-detection mechanism can employ a physical contact switch, or IR object sensing switch, which is actuated then the device is picked up off the countertop surface. Such mechanisms will become apparent in view of the teachings disclosed herein.

Modifications which Readily Come to Mind

In alternative embodiments of the present invention, illumination arrays **27**, **28** and **29** employed within the Multi-Mode Illumination Subsystem **14** may be realized using solid-state light sources other than LEDs, such as, for example, visible laser diode (VLDs) taught in great detail in WIPO Publication No. WO 02/43195 A2, published on May 30, 2002, assigned to Metrologic Instruments, Inc., and incorporated herein by reference in its entirety as if set forth fully herein. However, when using VLD-based illumination techniques in the Imaging-Based Bar Code Symbol Reader of the present invention, great care must be taken to eliminate or otherwise substantially reduce speckle-noise generated at the image detection array **22** when using coherent illumination source during object illumination and imaging operations. WIPO Publication No. WO 02/43195 A2, supra, provides diverse methods of and apparatus for eliminating or substantially reducing speckle-noise during image formation and detection when using VLD-based illumination arrays.

While CMOS image sensing array technology was described as being used in the preferred embodiments of the present invention, it is understood that in alternative embodiments, CCD-type image sensing array technology, as well as other kinds of image detection technology, can be used.

The bar code reader design described in great detail hereinabove can be readily adapted for use as an industrial or commercial fixed-position bar code reader/imager, having the interfaces commonly used in the industrial world, such as Ethernet TCP/IP for instance. By providing the system with an Ethernet TCP/IP port, a number of useful features will be enabled, such as, for example: multi-user access to such bar code reading systems over the Internet; control of multiple bar code reading system on the network from a single user application; efficient use of such bar code reading systems in live video operations; web-servicing of such bar code reading systems, i.e. controlling the system or a network of systems from an Internet Browser; and the like.

While the illustrative embodiments of the present invention have been described in connection with various types of bar code symbol reading applications involving 1-D and 2-D bar code structures, it is understood that the present invention can be use to read (i.e. recognize) any machine-readable indicia, dataform, or graphically-encoded form of intelligence, including, but not limited to bar code symbol structures, alphanumeric character recognition strings, handwriting, and diverse dataforms currently known in the art or to be developed in the future. Hereinafter, the term “code symbol” shall be deemed to include all, such information carrying structures and other forms of graphically-encoded intelligence.

Also, Imaging-Based Bar Code Symbol Readers of the present invention can also be used to capture and process various kinds of graphical images including photos and marks printed on driver licenses, permits, credit cards, debit cards, or the like, in diverse user applications.

It is understood that the image capture and processing technology employed in bar code symbol reading systems of the illustrative embodiments may be modified in a variety of ways which will become readily apparent to those skilled in the art of having the benefit of the novel teachings disclosed herein. All such modifications and variations of the illustrative embodiments thereof shall be deemed to be within the scope and spirit of the present invention as defined by the claims to Invention appended hereto.

What is claimed is:

1. An image formation system for detecting image light intensity reflected off an object placed in a field of view of a digital imaging based bar code symbol reading device, said system comprising:

an illumination array subsystem for producing a field of illumination upon said object during an image capture mode of said digital imaging based bar code symbol reading device

an image detection subsystem having an area type sensing array for detecting image light which is reflected off said object within the field of view of said digital imaging based bar code symbol reading device;

a photodiode for detecting image light intensity which is reflected off said object within said field of view digital imaging based bar code symbol reading device for subsequent processing by an automatic light exposure measurement and illumination control subsystem;

a beam splitter having a surface of a known reflection/transmission ratio and positioned within said field of view of said digital imaging based bar code symbol reading device whereby when said image light is reflected off said object in said field of view and upon said beam splitter a portion of said reflected image light is directed towards said area type sensing array during illumination operations in an image capture mode and a portion of said reflected image light is transmitted through said beam splitter and focused upon said photodiode, whereby said automatic light exposure measurement and illumination control subsystem controls at least one of the intensity or duration of illumination produced by said illumination array subsystem; and

a system control subsystem for activating and controlling said subsystem components described above.

2. The system of claim **1**, wherein illumination is collected from a center of said field of view of said digital imaging based bar code symbol reading device, detected by said photodiode and processed by said automatic light exposure measurement and illumination control system so as to generate a control signal for driving, at the proper intensity or for the proper duration, said illumination array, so that said area type image sensing array produces digital images of illuminated objects of sufficient brightness.

3. The system of claim **1**, wherein said illumination array subsystem includes light emitting diode based illumination.

4. The system of claim **3**, wherein once said area type image sensing array is activated by said system control subsystem, said system control subsystem automatically acti-

vates said automatic light exposure measurement and illumination control subsystem which, in response thereto, automatically drives said light emitting diode based illumination in a precise manner so as to illuminate the field of view of said digital imaging based bar code symbol reading device with said light emitting diode based illumination.

5. The system of claim **1**, wherein said beam splitter is positioned such that the optical axes of said image detection subsystem and said photodiode are exactly coincident thereby reducing one dimension of said digital imaging based bar code symbol reading device.

6. The system of claim **1**, wherein said beam splitter is a cube type beam splitter.

7. The system of claim **1**, wherein said beam splitter is a mirrored beam splitter.

* * * * *

BRITISH STANDARD

BS ISO/IEC 16022:2006

Incorporating Corrigenda October 2008 and February 2011

Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification

ICS 01.080.50; 35.040

EXHIBIT
1001

British Standards

National foreword

This British Standard is the UK implementation of ISO/IEC 16022:2006, incorporating corrigenda October 2008 and February 2011. It supersedes BS ISO/IEC 16022:2000 which is withdrawn.

The start and finish of text introduced or altered by corrigendum is indicated in the text by tags. Text altered by ISO/IEC corrigendum October 2008 is indicated in the text by AC1 AC1. Text altered by ISO/IEC corrigendum February 2011 is indicated in the text by AC2 AC2.

The UK participation in its preparation was entrusted by Technical Committee IST/34, Automatic identification and data capture techniques.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2006

ISBN 978 0 580 73412 0

Amendments/corrigenda issued since publication

Date	Comments
30 June 2009	Implementation of ISO/IEC corrigendum October 2008
30 June 2011	Implementation of ISO/IEC corrigendum February 2011

BS ISO/IEC 16022:2006
ISO/IEC 16022:2006 (E)

INTERNATIONAL STANDARD

ISO/IEC 16022

Second edition
2006-09-15

Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification

Technologies de l'information — Techniques d'identification automatique et de capture des données — Spécification de symbologie de code à barres Data Matrix

Reference number
ISO/IEC 16022:2006(E)

Contents

Page

Foreword ... vii
Introduction ... viii
1 Scope ... 1
2 Normative references ... 1
3 Terms, definitions, symbols and abbreviated terms, and mathematical/logical notations ... 2
3.1 Terms and definitions ... 2
3.2 Symbols and abbreviations ... 2
3.3 Mathematical/logical notations ... 3
4 Symbol description ... 3
4.1 Basic characteristics ... 3
4.2 Summary of additional features ... 4
4.3 Symbol structure ... 4
4.3.1 Finder pattern ... 5
4.3.2 Symbol sizes and capacities ... 5
5 ECC 200 requirements ... 5
5.1 Encode procedure overview ... 5
5.2 Data encodation ... 6
5.2.1 Overview ... 6
5.2.2 Default character interpretation ... 6
5.2.3 ASCII encodation ... 7
5.2.4 Symbology control characters ... 7
5.2.5 C40 encodation ... 9
5.2.6 Text encodation ... 11
5.2.7 ANSI X12 encodation ... 11
5.2.8 EDIFACT encodation ... 12
5.2.9 Base 256 encodation ... 12
5.3 User considerations ... 13
5.3.1 User selection of Extended Channel Interpretation ... 13
5.3.2 User selection of symbol size and shape ... 13
5.4 Extended Channel Interpretation ... 13
5.4.1 Encoding ECIs ... 14
5.4.2 ECIs and Structured Append ... 15
5.4.3 Post-decode protocol ... 15
5.5 ECC 200 symbol attributes ... 15
5.5.1 Symbol sizes and capacity ... 15
5.5.2 Insertion of Alignment Patterns into larger symbols ... 17
5.6 Structured Append ... 17
5.6.1 Basic principles ... 17
5.6.2 Symbol sequence indicator ... 17
5.6.3 File identification ... 18
5.6.4 FNC1 and Structured Append ... 18
5.6.5 Buffered and unbuffered operation ... 18
5.7 Error detection and correction ... 18
5.7.1 Reed-Solomon error correction ... 18
5.7.2 Generating the error correction codewords ... 18
5.7.3 Error correction capacity ... 19
5.8 Symbol construction ... 20
5.8.1 Symbol character placement ... 20
5.8.2 Alignment Pattern module placement ... 20

5.8.3 Finder Pattern module placement ... 20

6 ECC 000 - 140 requirements ... 21
6.1 Use recommendations ... 21
6.2 Encode procedure overview ... 21
6.3 Data encodation ... 21
6.3.1 Base 11 - Numeric encodation ... 23
6.3.2 Base 27 - Upper-case Alphabetic encodation ... 23
6.3.3 Base 37 - Upper-case Alphanumeric encodation ... 23
6.3.4 Base 41 - Upper-case Alphanumeric plus Punctuation encodation ... 24
6.3.5 ASCII encodation ... 24
6.3.6 8-bit byte encodation ... 24
6.4 User selection of error correction level ... 24
6.4.1 Selection of error correction level ... 24
6.4.2 Other error correction levels based on convolutional code algorithms ... 25
6.5 Constructing the Unprotected Bit Stream ... 25
6.5.1 Format ID Bit Field ... 25
6.5.2 CRC Bit Field ... 25
6.5.3 Data Length Bit Field ... 25
6.5.4 Data prefix construction ... 25
6.5.5 Completing the Unprotected Bit Stream ... 26
6.6 Constructing the Unrandomised Bit Stream ... 26
6.6.1 Header construction ... 26
6.6.2 Applying convolutional coding to create the Protected Bit Stream ... 26
6.6.3 Trailer construction ... 27
6.6.4 Completing the Unrandomised Bit Stream ... 27
6.7 Pattern randomising ... 27
6.8 Module placement in matrix ... 27

7 Symbol dimensions ... 27
7.1 Dimensions ... 27
7.2 Quiet zone ... 27

8 Symbol quality ... 27
8.1 Symbol quality parameters ... 28
8.1.1 Fixed pattern damage ... 28
8.1.2 Scan grade and overall symbol grade ... 28
8.1.3 Grid non-uniformity ... 28
8.2 Process control measurements ... 28

9 Reference decode algorithm for Data Matrix ... 28

10 User guidelines ... 38
10.1 Human readable interpretation ... 38
10.2 Autodiscrimination capability ... 38
10.3 System considerations ... 38

11 Transmitted data ... 38
11.1 Protocol for FNC1 (ECC 200 only) ... 38
11.2 Protocol for FNC1 in the second position (ECC 200 only) ... 38
11.3 Protocol for Macro characters in the first position (ECC 200 only) ... 38
11.4 Protocol for ECIs (ECC 200 only) ... 39
11.5 Symbology identifier ... 39
11.6 Transmitted data example ... 39

Annex A (normative) ECC 200 interleaving process ... 40
A.1 Schematic illustration ... 40
A.2 Starting sequence for interleaving in different sized symbols ... 40

Annex B (normative) ECC 200 pattern randomising ... 43
B.1 253-state algorithm ... 43
B.1.1 253-state randomising algorithm ... 43
B.1.2 253-state un-randomising algorithm ... 43
B.2 255-state algorithm ... 44

B.2.1 255-state randomising algorithm ... 44
B.2.2 255-state un-randomising algorithm ... 44

Annex C (normative) ECC 200 encodation character sets ... 45
C.1 C40 encodation character set ... 45
C.2 Text encodation character set ... 46
C.3 EDIFACT encodation character set ... 47

Annex D (normative) ECC 200 alignment patterns ... 48

Annex E (normative) ECC 200 Reed-Solomon error detection and correction ... 50
E.1 Error correction codeword generator polynomials ... 50
E.2 Error correction calculation ... 52
E.3 Calculation of error correction codewords ... 53

Annex F (normative) ECC 200 symbol character placement ... 55
F.1 Symbol character placement ... 55
F.2 Symbol character placement rules ... 57
F.2.1 Non-standard symbol character shapes ... 57
F.2.2 Symbol character arrangement ... 60
F.3 Symbol character placement examples for ECC 200 ... 63

Annex G (normative) ECC 000 - 140 symbol attributes ... 68
G.1 ECC 000 ... 68
G.2 ECC 050 ... 69
G.3 ECC 080 ... 70
G.4 ECC 100 ... 71
G.5 ECC 140 ... 72

Annex H (normative) ECC 000 - 140 data module placement grids ... 73

Annex I (normative) ECC 000 - 140 character encodation schemes ... 90
I.1 Base 11 encodation scheme ... 94
I.1.1 First stage procedure ... 94
I.1.2 Second stage procedure ... 94
I.1.3 Example ... 94
I.2 Base 27 encodation scheme ... 95
I.2.1 First stage procedure ... 95
I.2.2 Second stage procedure ... 95
I.2.3 Example ... 95
I.3 Base 37 encodation scheme ... 96
I.3.1 First stage procedure ... 96
I.3.2 Second stage procedure ... 96
I.3.3 Example ... 96
I.4 Base 41 encodation scheme ... 97
I.4.1 First stage procedure ... 97
I.4.2 Second stage procedure ... 97
I.4.3 Example ... 97

Annex J (normative) ECC 000 - 140 CRC algorithm ... 98
J.1 CRC state machine ... 98
J.2 CRC polynomial ... 98
J.3 CRC 2-byte header ... 98

Annex K (normative) ECC 000 - 140 error checking and correcting algorithms ... 100
K.1 ECC 000 ... 100
K.2 ECC 050 ... 100
K.3 ECC 080 ... 100
K.4 ECC 100 ... 100
K.5 ECC 140 ... 100
K.6 Processing the convolutional code ... 100
K.7 Convolutional codes reference decode algorithm ... 101

Annex L (normative) ECC 000 - 140 Master Random Bit Stream (in hexadecimal) ... 104

Annex M (normative) **Data Matrix print quality – symbology-specific aspects** **105**
M.1 Data Matrix Fixed Pattern Damage **105**
M.1.1 Features to be assessed **105**
M.1.2 Grading of the outside L of the fixed pattern **105**
M.1.3 Grading of the clock track and adjacent solid area segments **107**
M.1.4 Calculation and grading of average grade **111**
M.2 Scan grade **112**

Annex N (normative) **Symbology identifier** **113**

Annex O (informative) **ECC 200 encode example** **114**

Annex P (informative) **Encoding data using the minimum symbol data characters for ECC 200** **116**

Annex Q (informative) **ECC 000 - 140 encode example using ECC 050** **120**
Q.1 Encode example **120**
Q.2 CRC calculation for example **125**

Annex R (informative) **Useful process control techniques** **128**
R.1 Symbol contrast **128**
R.2 Special reference symbol **128**
R.3 Assessing Axial Nonuniformity **129**
R.4 Visual inspection for symbol distortion and defects **129**

Annex S (informative) **Autodiscrimination capability** **130**

Annex T (informative) **System considerations** **131**

Bibliography **132**

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 16022 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 16022:2000), which has been technically revised. It also incorporates the Technical Corrigendum ISO/IEC 16022:2000/Cor.1:2004.

Introduction

Data Matrix is a two-dimensional matrix symbology which is made up of nominally square modules arranged within a perimeter finder pattern. Though primarily shown and described in this International Standard as a dark symbol on light background, Data Matrix symbols can also be printed to appear as light on dark.

Manufacturers of bar code equipment and users of the technology require publicly available standard symbology specifications to which they can refer when developing equipment and application standards. The publication of standardised symbology specifications is designed to achieve this.

Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification

1 Scope

This International Standard defines the requirements for the symbology known as Data Matrix. It specifies the Data Matrix symbology characteristics, data character encodation, symbol formats, dimensions and print quality requirements, error correction rules, decoding algorithm, and user-selectable application parameters.

It applies to all Data Matrix symbols produced by any printing or marking technology.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15424, *Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)*

ISO/IEC 19762-1, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 1: General terms relating to AIDC*

ISO/IEC 19762-2, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 2: Optically readable media (ORM)*

ISO/IEC 15415, *Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Two-dimensional symbols*

ISO/IEC 15416, *Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Linear symbols*

ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange*

ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

ISO/IEC 8859-5:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 5: Latin/Cyrillic alphabet*

AIM Inc. ITS/04-001 International Technical Standard: *Extended Channel Interpretations — Part 1: Identification Schemes and Protocol*

3 Terms, definitions, symbols and mathematical/logical notations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762-1, ISO/IEC 19762-2 and the following apply.

3.1.1
codeword
symbol character value, an intermediate level of coding between source data and the graphical encodation in the symbol

3.1.2
module
single cell in a matrix symbology used to encode one bit of data, nominally a square shape in Data Matrix

3.1.3
convolutional coding
error checking and correcting (ECC) algorithm that processes a set of input bits into a set of output bits that can recover from damage by breaking the input bits into blocks, then convolving each input block with the contents of a multi-stage shift register to produce protected output blocks

NOTE These encoders can be constructed in hardware using input and output switches, shift registers, and exclusive-or (XOR) gates.

3.1.4
pattern randomising
procedure to convert an original bit pattern to another bit pattern, intended to reduce the probability of repeating patterns occurring in the symbol, by inverting selected bits

3.2 Symbols

For the purposes of this document, the following mathematical symbols apply unless defined locally.

d number of error correction codewords

e number of erasures

k (for ECC 000 - 140) the number of bits in a complete segment input to the state machine to generate the convolutional code (for ECC 200) total number of error correction codewords

m the memory order of the convolutional code

n (for ECC 000 - 140) the number of bits in a complete segment generated by the state machine producing the convolutional code (for ECC 200) total number of data codewords

N the numerical base in an encodation scheme

p number of codewords reserved for error detection

S symbol character

t number of errors

u the input bit segment to the state machine, taken k bits at a time

v the output bit segment from the state machine, generated n bits at a time

X horizontal and vertical width of a module

ε error correction codeword

3.3 Mathematical/logical notations

For the purposes of this document, the following notations and mathematical operations apply.

div integer division operator

mod integer remainder after division

XOR exclusive-or logic function whose output is one only when its two inputs are not equivalent.

LSB least significant bit

MSB most significant bit

4 Symbol description

4.1 Basic characteristics

Data Matrix is a two-dimensional matrix symbology.

There are two types:

ECC 200 which uses Reed-Solomon error correction. ECC 200 is recommended for new applications.

ECC 000 - 140 with several available levels of convolutional error correction, referred to as ECC 000, ECC 050, ECC 080, ECC 100 and ECC 140 respectively. ECC 000 - 140 should only be used in closed applications where a single party controls both the production and reading of the symbols and is responsible for overall system performance.

The characteristics of Data Matrix are:

a) Encodable character set:

1) values 0 – 127 in accordance with the US national version of ISO/IEC 646

NOTE 1 This version consists of the G0 set of ISO/IEC 646 and the C0 set of ISO/IEC 6429 with values 28 – 31 modified to FS, GS, RS and US respectively.

2) values 128 - 255 in accordance with ISO 8859-1. These are referred to as extended ASCII.

b) Representation of data: A dark module is a binary one and a light module is a zero.

NOTE 2 This International Standard specifies Data Matrix symbols in terms of dark modules marked on a light background. However, subclause 4.2 provides that symbols may also be produced with light and dark modules reversed in colour (see 4.2), and in such symbols references in this International Standard to dark modules should be taken as references to light modules, and vice versa.

c) Symbol size in modules (not including quiet zone):

ECC 200 10 x 10 to 144 x 144 even values only

ECC 000 – 140 9 x 9 to 49 x 49, odd values only

d) Data characters per symbol (for maximum symbol size in ECC200):

1) Alphanumeric data: up to 2 335 characters

2) 8-bit byte data: 1 555 characters

3) Numeric data: 3 116 digits.

e) Selectable error correction:

ECC 200: Reed-Solomon error correction.

ECC 000 - 140: Four levels of convolutional error correction, plus the option to apply only error detection

f) Code type: Matrix

g) Orientation independence: Yes

4.2 Summary of additional features

The following summarises additional features which are inherent or optional in Data Matrix:

a) Reflectance reversal: (Inherent): Symbols are intended to be read when marked so that the image is either dark on light or light on dark (see Figure 1). The specifications in this International Standard are based on dark images on a light background, therefore references to dark or light modules should be taken as references to light or dark modules respectively in the case of symbols produced with reflectance reversal.

b) Extended Channel Interpretations: (ECC 200 only, optional): This mechanism enables characters from other character sets (e.g. Arabic, Cyrillic, Greek, Hebrew) and other data interpretations or industry-specific requirements to be represented.

c) Rectangular symbols: (ECC 200 only, optional): Six symbol formats are specified in a rectangular form.

d) Structured append: (ECC 200 only, optional): This allows files of data to be represented in up to 16 Data Matrix symbols. The original data can be correctly reconstructed regardless of the order in which the symbols are scanned.

4.3 Symbol structure

Each Data Matrix symbol consists of data regions which contain nominally square modules set out in a regular array. In larger ECC 200 symbols, data regions are separated by alignment patterns. The data region, or set of data regions and alignment patterns, is surrounded by a finder pattern, and this shall in turn be surrounded on all four sides by a quiet zone border. Figure 1 illustrates an ECC 140 and two representations of an ECC 200 symbol.

(a) ECC200, dark on light (b) ECC200, light on dark (c) ECC140, dark on light

Figure 1 — ECC 200 (a & b) and ECC 140 (c) encoding "A1B2C3D4E5F6G7H8I9J0K1L2"

4.3.1 Finder pattern

The finder pattern is a perimeter to the data region and is one module wide. Two adjacent sides, the left and lower sides, forming the L boundary, are solid dark lines; these are used primarily to determine physical size, orientation and symbol distortion. The two opposite sides are made up of alternating dark and light modules. These are used primarily to define the cell structure of the symbol, but also can assist in determining physical size and distortion. The extent of the quiet zone is indicated by the corner marks in Figure 1.

4.3.2 Symbol sizes and capacities

ECC 200 symbols have an even number of rows and an even number of columns. Some symbols are square with sizes from 10 x 10 to 144 x 144 not including quiet zones. Some symbols are rectangular with sizes from 8 x 18 to 16 x 48 not including quiet zones. All ECC 200 symbols can be recognised by the upper right corner module being light. The complete attributes of ECC 200 symbols are given in Table 7 in Section 5.5.

ECC 000 - 140 symbols have an odd number of rows and an odd number of columns. Symbols are square with sizes from 9 x 9 to 49 x 49 (modules) not including quiet zones. These symbols can be recognised by the upper right corner module being dark. The complete attributes of ECC 000 - 140 symbols are given in Annex G.

5 ECC 200 requirements

5.1 Encode procedure overview

This section provides an overview of the encoding procedure. Following sections will provide more details. An encoding example for ECC 200 is given in Annex O. The following steps convert user data to an ECC 200 symbol:

Step 1: Data encodation

> Analyse the data stream to identify the variety of different characters to be encoded. ECC 200 includes various encodation schemes which allow a defined set of characters to be converted into codewords more efficiently than the default scheme. Insert additional codewords to switch between the encodation schemes and to perform other functions. Add pad characters as needed to fill the required number of codewords. If the user does not specify the matrix size, then choose the smallest size that accommodates the data. A complete list of matrix sizes is shown in Section 5.5, Table 7.

Table 1 — Encodation schemes for ECC 200

Encodation scheme	Characters	Bits per data character
ASCII	double digit numerics	4
	ASCII values 0 - 127	8
	Extended ASCII values 128 - 255	16
C40	Upper-case alphanumeric	5,33
	Lower case and special characters	10,66[a]
Text	Lower-case alphanumeric	5,33
	Upper case and special characters	10,66[b]
X12	ANSI X12 EDI data set	5,33
EDIFACT	ASCII values 32 - 94	6
Base 256	All byte values 0 - 255	8

[a] encoded as two C40 values as result of use of a shift character

[b] encoded as two Text values as result of use of a shift character

Step 2: Error checking and correcting codeword generation

For symbols with more than 255 codewords, sub-divide the codeword stream into interleaved blocks to enable the error correction algorithms to be processed as shown in Annex A. Generate the error correction codewords for each block. The result of this process expands the codeword stream by the number of error correction codewords. Place the error correction codewords after the data codewords.

Step 3: Module placement in matrix

Place the codeword modules in the matrix. Insert the alignment pattern modules, if any, in the matrix. Add the finder pattern modules around the matrix.

5.2 Data encodation

5.2.1 Overview

The data may be encoded using any combination of six encodation schemes (see Table 1). ASCII encodation is the basic scheme. All other encodation schemes are invoked from ASCII encodation and return to this scheme. The compaction efficiencies given in Table 1 need to be interpreted carefully. The best scheme for a given set of data may not be the one with the fewest bits per data character. If the highest degree of compaction is required, account has to be taken of switching between encodation schemes and between code sets within an encodation scheme (see Annex P). It should also be noted that even if the number of codewords is minimised, the codeword stream might need to be expanded to fill a symbol. This fill process is done using pad characters.

5.2.2 Default character interpretation

The default character interpretation for character values 0 to 127 shall conform to ISO/IEC 646. The default character interpretation for character values 128 to 255 shall conform to ISO 8859-1: Latin Alphabet No. 1. The graphical representation of data characters shown throughout this document complies with the default interpretation. This interpretation can be changed using Extended Channel Interpretation (ECI) escape sequences, see 5.4. The default interpretation corresponds to ECI 000003.

5.2.3 ASCII encodation

ASCII encodation is the default set for the first symbol character in all symbol sizes. It encodes ASCII data, double density numeric data and symbology control characters. Symbology control characters include function characters, the pad character and the switches to other code sets. ASCII data is encoded as codewords 1 to 128 (ASCII value plus 1). Extended ASCII (data values 128 to 255) is encoded using the upper shift symbology control character (see 5.2.4.2). The digit pairs 00 to 99 are encoded with codewords 130 to 229 (numeric value plus 130). The ASCII code assignments are shown in Table 2.

Table 2 — ASCII encodation values

Codeword	Data or function
1 - 128	ASCII data (ASCII value + 1)
129	Pad
130 - 229	2-digit data 00 - 99 (Numeric Value + 130)
230	Latch to C40 encodation
231	Latch to Base 256 encodation
232	FNC1
233	Structured Append
234	Reader Programming
235	Upper Shift (shift to Extended ASCII)
236	05 Macro
237	06 Macro
238	Latch to ANSI X12 encodation
239	Latch to Text encodation
240	Latch to EDIFACT encodation
241	ECI Character
242 - 255	Not to be used in ASCII encodation

5.2.4 Symbology control characters

ECC 200 symbols have several special symbology control characters, which have particular significance to the encodation scheme. These characters shall be used to instruct the decoder to perform certain functions or to send specific data to the host computer as described in 5.2.4.1 to 5.2.4.9. These symbology control characters, with the exception of values from 242 through 255, are found in the ASCII encodation set (see Table 2).

5.2.4.1 Latch characters

A Latch Character shall be used to switch from ASCII encodation to one of the other encodation schemes. All codewords which follow a Latch Character shall be compacted according to the new encodation scheme. The encodation schemes have different methods for returning to the ASCII encodation set.

5.2.4.2 Upper Shift character

The Upper Shift character is used in combination with an ASCII value (1 - 128) to encode an extended ASCII character (129-255). An extended ASCII character encoded in the ASCII, C40, or Text encodation scheme requires a preceding Upper Shift character and the extended ASCII character value decreased by 128 is then encoded according to the rules of the encodation scheme. In ASCII encodation, the Upper Shift character is represented by codeword 235. The reduced data value (i.e. ASCII value minus 128) is transformed into its codeword value by adding 1. For example, to encode ¥ (Yen currency symbol) (ASCII value 165), an Upper Shift character (Codeword 235) is followed by value 37 (165 - 128), which is encoded as codeword 38. If there are long data strings of characters from the extended ASCII range, a Latch to Base 256 encodation should be more efficient.

5.2.4.3 Pad character

If the encoded data, irrespective of the encodation scheme in force, does not fill the data capacity of the symbol, pad characters (value 129 in ASCII encodation) shall be added to fill the remaining data capacity of the symbol. The pad characters shall only be used for this purpose. Before inserting pad characters, it is necessary to return to ASCII encodation if in any other encodation mode. The 253-State pattern randomising algorithm is applied to the pad characters starting at the second pad character and continuing to the end of the symbol (see Annex B.1).

5.2.4.4 Extended Channel Interpretation character

An Extended Channel Interpretation (ECI) character is used to change from the default interpretation used to encode data. The Extended Channel Interpretation protocol is common across a number of symbologies and its application to ECC 200 is defined more fully in 5.4. The ECI character shall be followed by one, two, or three codewords which identify the ECI being invoked. The new ECI remains in place until the end of the encoded data, or until another ECI character is used to invoke another interpretation.

5.2.4.5 Shift characters in C40 and Text encodation

In C40 and Text Encoding, three special characters, called shift characters, are used as a prefix to one of 40 values to encode about three quarters of the ASCII characters. This allows the remaining ASCII characters to be encoded in a more condensed way with single values.

5.2.4.6 FNC1 alternate data type identifier

To encode data to conform to specific industry standards as authorised by AIM Inc., a FNC1 character shall appear in the first or second symbol character position (or in the fifth or sixth data positions of the first symbol of Structured Append). FNC1 encoded in any other position is used as a field separator and shall be transmitted as $^{G}{}_{S}$ control character (ASCII value 29).

5.2.4.7 Macro characters

Data Matrix provides a means of abbreviating an industry specific header and trailer in one symbol character. This feature exists to reduce the number of symbol characters needed to encode data in a symbol using certain structured formats. A Macro character must be in the first character position of a symbol. They shall not be used in conjunction with Structured Append and their functions are defined in Table 3. The header shall be transmitted as a prefix to the data stream and the trailer shall be transmitted as a suffix to the data stream. The symbology identifier, if used, shall precede the header.

Table 3 — Macro functions

Macro codeword	Name	Interpretation	
		Header	Trailer
236	05 Macro	[)>$^{R}_{S}$05$^{G}_{S}$	$^{R}_{S}{}^{E}O_{T}$
237	06 Macro	[)>$^{R}_{S}$06$^{G}_{S}$	$^{R}_{S}{}^{E}O_{T}$

5.2.4.8 Structured Append character

A Structured Append character is used to indicate that the symbol is part of a Structured Append sequence according to the rules defined in 5.6.

5.2.4.9 Reader Programming character

A Reader Programming character indicates that the symbol encodes a message used to program the reader system. The Reader Programming character shall appear as the first codeword of the symbol and Reader Programming shall not be used with Structured Append.

5.2.5 C40 encodation

The C40 encodation scheme is designed to optimise the encoding of upper-case alphabetic and numeric characters but also enables other characters to be encoded by the use of shift characters in conjunction with the data character.

C40 characters are partitioned into 4 subsets. Characters of the first set, called the basic set, are the three special shift characters, the space character, and the ASCII characters A-Z and 0-9. They are assigned to a single C40 values. Characters of the other sets are assigned to one of the three shift characters, pointing to one of the 3 remaining subset, followed by one of the C40 values (see Annex C, Table C.1).

As a first stage, each data character is converted into a single C40 value or a pair of C40 values. The complete string of C40 values is then decomposed into groups of three values (special rules apply if one or two values remain at the end, see 5.2.5.2.). Each triplet (C1, C2, C3) is then encoded into a 16-bit value according to the formula: (1600 * C1) + (40 * C2) + C3 + 1. Each 16-bit value is then separated into 2 codewords by taking the most significant 8 bits and the least significant 8 bits.

5.2.5.1 Switching to and from C40 encodation

It is possible to switch to C40 encodation from ASCII encodation using the appropriate latch codeword (230). Codeword 254 immediately following a pair of codewords in C40 encodation acts as an Unlatch codeword to switch back to ASCII encodation. Otherwise, the C40 encodation remains in effect to the end of the data encoded in the symbol.

5.2.5.2 C40 encodation rules

Each pair of codewords represents a 16-bit value where the first codeword represents the most significant 8 bits. Three C40 values ($C1$, $C2$, $C3$) shall be encoded as:

$$(1600 * C1) + (40 * C2) + C3 + 1$$

which produces a value from 1 to 64000. Figure 2 illustrates three C40 values compacted into two codewords. Characters in the Shift 1, Shift 2 and Shift 3 sets shall be encoded by first encoding the appropriate shift character, and then the C40 value for the data. C40 encodation may be in effect at the end of the symbol's codewords which encode data.

The following rules apply when only one or two symbol characters remain in the symbol before the start of the error correction codewords:

a) If two symbol characters remain and three C40 values remain to be encoded (which may include both data and shift characters) encode the three C40 values in the last two symbol characters. A final Unlatch codeword is not required.

b) If two symbol characters remain and two C40 values remain to be encoded (the first C40 value may be a shift or data character but the second must represent a data character); encode the two remaining C40 values followed by a pad C40 value of 0 (Shift 1) in the last two symbol characters. A final Unlatch codeword again is not required.

c) If two symbol characters remain and only one C40 value (data character) remains to be encoded, the first symbol character is encoded as an Unlatch character and the last symbol character is encoded with the data character using the ASCII encodation scheme.

d) If one symbol character remains and one C40 value (data character) remains to be encoded, the last symbol character is encoded with the data character using the ASCII encodation scheme. The Unlatch character is not encoded, but is assumed, before the last symbol character.

   In all other cases, either an Unlatch character is used to exit the C40 encodation scheme before the end of the symbol, or a larger symbol size is required to encode the data.

Data characters	AIM
C40 values	14, 22, 26
Calculate 16-bit value	(1600 * 14) + (40 * 22) + 26 + 1 = 23307
1st codeword: (16-bit value) div 256	23307 div 256 = 91
2nd codeword: (16-bit value) mod 256	23307 mod 256 = 11
Codewords	91, 11

Figure 2 — Example of C40 encoding

5.2.5.3 Use of Upper Shift with C40

In C40 encodation the Upper Shift character is not a symbology function character but a shift within the encodation set. When a data character from the extended ASCII character range is encountered, three or four values in C40 encodation need to be encoded according to the following rule:

IF [ASCII value - 128] is in the Basic Set then:

[1(Shift 2)] [30(Upper Shift)] [V(ASCII value - 128)]

ELSE

[1(Shift 2)] [30(Upper Shift)] [0, 1, or 2(Shift 1, 2, or 3)] [V(ASCII value - 128)]

In the rule the number in [ ] equates to the C40 values from Annex C.1; V has been used to indicate the appropriate C40 value.

5.2.6 Text encodation

Text encodation is designed to encode normal printed text, which is predominantly lowercase characters. It is similar in structure to the C40 encodation set, except that lowercase alphabetic characters are directly encoded (i.e. without using a shift). Upper-case alphabetic characters are preceded by a Shift 3. The full Text encodation character set assignments are shown in Annex C, Table C.2.

5.2.6.1 Switching to and from Text encodation

It is possible to switch to Text encodation from ASCII encodation using the appropriate latch codeword (239). Codeword 254 immediately following a pair of codewords in text encodation acts as an Unlatch codeword to switch back to ASCII encodation. Otherwise, the Text encodation remains in effect to the end of the data encoded in the symbol.

5.2.6.2 Text encodation rules

The rules for C40 encodation apply.

5.2.7 ANSI X12 encodation

ANSI X12 encodation is used to encode the standard ANSI X12 electronic data interchange characters, which are compacted three data characters to two codewords in a manner similar to C40 encodation. It encodes upper-case alphabetic characters, numerics, space and the three standard ANSI X12 terminator and separator characters. The ANSI X12 code assignments are shown in Table 4. There are no shift characters in the ANSI X12 encodation set.

Table 4 — ANSI X12 encodation set

X12 value	Encoded characters	ASCII values
0	X12 segment terminator <CR>	13
1	X12 segment separator *	42
2	X12 sub-element separator >	62
3	space	32
4 - 13	0 - 9	48 - 57
14 - 39	A - Z	65 - 90

5.2.7.1 Switching to and from ANSI X12 encodation

It is possible to switch to ANSI X12 encodation from ASCII encodation using the appropriate latch codeword (238). Codeword 254 immediately following a pair of codewords in ANSI X12 encodation acts as an Unlatch codeword to switch back to ASCII encodation. Otherwise, the ANSI X12 encodation remains in effect to the end of the data encoded in the symbol.

5.2.7.2 ANSI X12 encodation rules

The rules of C40 encodation apply. The exception is at the end of encoding ANSI X12 data. If the data characters do not fully utilise pairs of codewords, then following the last complete pair of codewords switch to ASCII using codeword 254 and continue using ASCII encodation, except when a single symbol character is left at the end before the first error correction character. This single symbol character uses the ASCII encodation scheme without requiring an Unlatch codeword.

5.2.8 EDIFACT encodation

The EDIFACT encodation scheme includes 63 ASCII values (values from 32 to 94) plus an Unlatch character (binary 011111) to return to ASCII encodation. EDIFACT encodation encodes four data characters in three codewords. It includes all the numeric, alphabetic and punctuation characters defined in the EDIFACT Level A character set without any of the shifts required in C40 encodation.

5.2.8.1 Switching to and from EDIFACT encodation

It is possible to switch to EDIFACT encodation from ASCII encodation using the appropriate latch codeword (240). The Unlatch character in EDIFACT encodation shall be used as a terminator at the end of EDIFACT encodation, which reverts to ASCII encodation.

5.2.8.2 EDIFACT encodation Rules

The EDIFACT encodation character set is defined in Annex C, Table C.3. There is a simple relationship between the 6-bit EDIFACT value and the ASCII 8-bit byte. The leading two bits of the 8-bit byte are ignored to create the EDIFACT 6-bit value, as illustrated in Figure 3. Strings of four EDIFACT characters are encoded in three codewords. For a simple encodation process, the leading two bits of the 8-bit byte are removed. The remaining 6-bit byte is the EDIFACT value and shall be directly encoded into the codeword as illustrated in Figure 4. When EDIFACT encodation is terminated with the Unlatch character, any remaining bits left in the single symbol character shall be filled with zeros. ASCII mode starts with the next symbol character. If EDIFACT encodation is in effect at the end of the symbol before the first error correction character, and only one or two codewords remain after the last EDIFACT codeword triplet, these remaining codewords shall be encoded in ASCII encodation without requiring an Unlatch character.

Data character	ASCII		EDIFACT value
	Decimal value	8-bit binary value	
A	65	01000001	000001
9	57	00111001	111001
NOTE During the decode process, if the leading (6th) bit is 1, the bits 00 are prefixed to create the 8-bit byte. If the leading (6th) bit is 0, the bits 01 are prefixed to create the 8-bit byte. The exception to this is the EDIFACT value 011111 which is the symbology control Unlatch character to return to ASCII encodation.			

Figure 3 — The relationship between the EDIFACT value and the 8-bit byte value

Data characters	D			A			T			A		
Binary values (Table C.3)	00	01	00	00	00	01	01	01	00	00	00	01
Divide into 3 8-bit bytes	00	01	00	00	00	01	01	01	00	00	00	01
Codeword values	16				21				1			

Figure 4 — Example of EDIFACT encodation

5.2.9 Base 256 encodation

The Base 256 encodation scheme shall be used to encode any 8-bit byte data, including extended channel interpretations and binary data. The default interpretation is defined in 5.2.2. The 255-State pattern randomising algorithm is applied to each Base 256 sequence within the encoded data (see B.2). It starts after the latch to Base 256 encodation and ends at the last character specified by the Base 256 field length.

5.2.9.1 Switching to and from Base 256 encodation

It is possible to switch to Base 256 encodation from ASCII encodation using the appropriate latch codeword (231). At the end of Base 256 encodation, encodation automatically reverts to ASCII encodation. The appropriate ECI, if other than the default, shall be invoked prior to switching. The ECI sequence need not occur immediately before switching to Base 256 encodation.

5.2.9.2 Base 256 encodation rules

After switching to Base 256 encodation, the first one (*d1*) or two (*d1*, *d2*) codewords define the data field length in bytes. Table 5 specifies how the field length is defined. Thereafter, all encodation shall be of the byte values.

Table 5 — Base 256 field length

Field Length	Values of *d1*, *d2*	Permitted Values of d
Remainder of Symbol	*d1* = 0	*d1* = 0
1 to 249	*d1* = length	*d1* = 1 to 249
250 to 1555	*d1* = (length DIV 250) + 249	*d1* = 250 to 255
	d2 = length MOD 250	*d2* = 0 to 249

5.3 User considerations

ECC 200 offers flexibility in the way data is encoded. Alternate character sets may be invoked using the ECI protocol. Data may be encoded in square or rectangular symbols. Where the message length exceeds the capacity of a single symbol, it is also possible to encode it in a Structured Append sequence of up to 16 separate but logically linked ECC 200 symbols (see 5.6).

5.3.1 User selection of Extended Channel Interpretation

The use of an alternative Extended Channel Interpretation to identify a particular code page or more specific data interpretation requires additional codewords to invoke the feature. The use of the Extended Channel Interpretation protocol (see 5.4) provides the capability to encode data from alphabets other than the Latin alphabet (ISO 8859-1 Latin Alphabet No. 1) supported by the default interpretation (ECI 000003).

5.3.2 User selection of symbol size and shape

ECC 200 has twenty-four square and six rectangular symbol configurations. The size and shape may be selected to suit the requirement of the application. These configurations are technically specified in 5.5.

5.4 Extended Channel Interpretation

The Extended Channel Interpretation (ECI) protocol allows the output data stream to have interpretations different from that of the default character set. The ECI protocol is defined consistently across a number of symbologies. Four broad types of interpretations are supported in Data Matrix:

a) international character sets (or code pages)

b) general purpose interpretations such as encryption and compaction

c) user defined interpretations for closed systems

d) control information for structured append in unbuffered mode.

The Extended Channel Interpretation protocol is fully specified in AIM Inc. International Technical Specification – Extended Channel Interpretations Part 1. The protocol provides a consistent method to specify particular interpretations on byte values before printing and after decoding. The Extended Channel Interpretation is identified by a 6-digit number which is encoded in the Data Matrix symbol by the ECI character followed by one to three codewords. Specific interpretations are listed in AIM Inc. Extended Channel Interpretations Character Set Register. The Extended Channel Interpretation can only be used with readers enabled to transmit the symbology identifiers. Readers that are not enabled to transmit the symbology identifier shall not transmit the data from any symbol containing an ECI. An exception can be made if the ECI(s) can be handled entirely within the reader.

The Extended Channel Interpretation protocol shall only be applied to ECC 200 symbols. A specified Extended Channel Interpretation may be invoked anywhere in the encoded message.

5.4.1 Encoding ECIs

The various encodation schemes of Data Matrix for ECC 200 (defined in Table 1) may be applied under any of the Extended Channel Interpretations. The ECI can only be invoked from ASCII encodation; once this has occurred, switching may take place between any of the encodation schemes. The encodation mode used is determined strictly by the 8-bit data values being encoded and does not depend on the Extended Channel Interpretation in force. For example, a sequence of values in the range 48 to 57 (decimal) would be most efficiently encoded in numeric mode even if they were not to be interpreted as numbers. The ECI assignment is invoked using codeword 241 (ECI character) in ASCII encodation. One, two, or three additional codewords are used to encode the ECI Assignment number. The encodation rules are defined in Table 6.

The following examples illustrate the encodation:

<u>ECI = 015000</u>

Codewords:

[241] [(15000 - 127) div 254 + 128] [(15000 - 127) mod 254 + 1]

= [241] [58 + 128] [141 + 1]

= [241] [186] [142]

<u>ECI = 090000</u>

Codewords:

[241] [(90000 - 16383) div 64516 + 192] [((90000 - 16383) div 254) mod 254 + 1] [(90000 - 16383) mod 254 + 1]

= [241] [1 + 192] [289 mod 254 + 1] [211 + 1]

= [241] [193] [36] [212]

Table 6 — Encoding ECI assignment numbers in ECC 200

ECI assignment value	Codeword sequence	Codeword values	Ranges
000000 to 000126	C_0	241	
	C_1	*ECI_no* + 1	C_1 = (1 to 127)
000127 to 016382	C_0	241	
	C_1	(*ECI_no* - 127) div 254 + 128	C_1 = (128 to 191)
	C_2	(*ECI_no* - 127) mod 254 + 1	C_2 = (1 to 254)
0016383 to 999999	C_0	241	
	C_1	(*ECI_no* - 16383) div 64516 +192	C_1 = (192 to 207)
	C_2	[(*ECI_no* - 16383) div 254] mod 254 + 1	C_2 = (1 to 254)
	C_3	(*ECI_no* - 16383) mod 254 + 1	C_3 = (1 to 254)

5.4.2 ECIs and Structured Append

ECIs may occur anywhere in the message encoded in a single or Structured Append (see 5.6) set of Data Matrix symbols. Any ECI invoked shall apply until the end of the encoded data, or until another ECI is encountered. Thus the interpretation of the ECI may straddle two or more symbols.

5.4.3 Post-decode protocol

The protocol for transmitting ECI data shall be as defined in 11.4. When using ECIs, symbology identifiers (see 11.5) shall be fully implemented and the appropriate symbology identifier transmitted as a preamble.

5.5 ECC 200 symbol attributes

5.5.1 Symbol sizes and capacity

There are 24 square symbols and 6 rectangular symbols available in ECC 200. These are as specified in Table 7.

Table 7 — ECC 200 symbol attributes

Symbol size[a]		Data region		Mapping matrix size	Total codewords		Reed-Solomon block		Inter–leaved blocks	Maximum data capacity			% of codewords used for error correction	Max. correctable codewords
Row	Col	Size	No.		Data	Error	Data	Error		Num.	Alphanum.[d]	Byte		Error/ erasure[b]
10	10	8 x 8	1	8 x 8	3	5	3	5	1	6	3	1	62,5	2/0
12	12	10 x 10	1	10 x 10	5	7	5	7	1	10	6	3	58,3	3/0
14	14	12 x 12	1	12 x 12	8	10	8	10	1	16	10	6	55,6	5/7
16	16	14 x 14	1	14 x 14	12	12	12	12	1	24	16	10	50	6/9
18	18	16 x 16	1	16 x 16	18	14	18	14	1	36	25	16	43,8	7/11
20	20	18 x 18	1	18 x 18	22	18	22	18	1	44	31	20	45	9/15
22	22	20 x 20	1	20 x 20	30	20	30	20	1	60	43	28	40	10/17
24	24	22 x 22	1	22 x 22	36	24	36	24	1	72	52	34	40	12/21
26	26	24 x 24	1	24 x 24	44	28	44	28	1	88	64	42	38,9	14/25
32	32	14 x 14	4	28 x 28	62	36	62	36	1	124	91	60	36,7	18/33
36	36	16 x 16	4	32 x 32	86	42	86	42	1	172	127	84	32,8	21/39
40	40	18 x 18	4	36 x 36	114	48	114	48	1	228	169	112	29,6	24/45
44	44	20 x 20	4	40 x 40	144	56	144	56	1	288	214	142	28	28/53
48	48	22 x 22	4	44 x 44	174	68	174	68	1	348	259	172	28,1	34/65
52	52	24 x 24	4	48 x 48	204	84	102	42	2	408	304	202	29,2	42/78
64	64	14 x 14	16	56 x 56	280	112	140	56	2	560	418	277	28,6	56/106
72	72	16 x 16	16	64 x 64	368	144	92	36	4	736	550	365	28,1	72/132
80	80	18 x 18	16	72 x 72	456	192	114	48	4	912	682	453	29,6	96/180
88	88	20 x 20	16	80 x 80	576	224	144	56	4	1 152	862	573	28	112/212
96	96	22 x 22	16	88 x 88	696	272	174	68	4	1 392	1 042	693	28,1	136/260
104	104	24 x 24	16	96 x 96	816	336	136	56	6	1 632	1 222	813	29,2	168/318
120	120	18 x 18	36	108 x 108	1 050	408	175	68	6	2 100	1 573	1 047	28	204/390
132	132	20 x 20	36	120 x 120	1 304	496	163	62	8	2 608	1 954	1 301	27,6	248/472
144	144	22 x 22	36	132 x 132	1 558	620	156	62	8[c]	3 116	2 335	1 555	28,5	310/590
							155	62	2[c]					
Rectangular Symbols														
8	18	6 x 16	1	6 x 16	5	7	5	7	1	10	6	3	58,3	3/0
8	32	6 x 14	2	6 x 28	10	11	10	11	1	20	13	8	52,4	5/0
12	26	10 x 24	1	10 x 24	16	14	16	14	1	32	22	14	46,7	7/11
12	36	10 x 16	2	10 x 32	22	18	22	18	1	44	31	20	45,0	9/15
16	36	14 x 16	2	14 x 32	32	24	32	24	1	64	46	30	42,9	12/21
16	48	14 x 22	2	14 x 44	49	28	49	28	1	98	72	47	36,4	14/25

a symbol size does not include quiet zones

b See 5.7.3

c In the largest symbol (144 x 144), the first eight Reed-Solomon blocks are 218 codewords long encoding 156 data codewords, and the last two blocks encode 217 codewords (155 data codewords). All the blocks have 62 error correction codewords.

d Based on text or C40 encoding without switching or shifting; for other encoding schemes, this value may vary depending on the mix and grouping of character sets

5.5.2 Insertion of Alignment Patterns into larger symbols

As shown in Table 7, square symbols 32 x 32 and larger and four rectangular symbols (8 x 32, 12 x 36, 16 x 36, and 16 x 48) have two or more data regions. These data regions are bounded by alignment patterns (see Annex D). The square symbols are divided into 4, 16, or 36 data regions (as illustrated in Annex D, Figures D.1, D.2, and D.3). The rectangular symbols are divided into two data regions (as illustrated in Annex D, Figure D.4). The alternating dark modules of the alignment pattern shall be to the top and right of a data region and identify the even columns and rows.

5.6 Structured Append

5.6.1 Basic principles

Up to 16 ECC 200 symbols may be appended in a structured format. If a symbol is part of a Structured Append, this is indicated by codeword 233 in the first symbol character position. This is immediately followed by three structured append codewords. The first codeword is the symbol sequence indicator. The second and third codewords are the file identification.

5.6.2 Symbol sequence indicator

This codeword indicates the position of the symbol within the set (up to 16) of ECC 200 symbols in the Structured Append format in the form *m* of *n* symbols. The first 4 bits of this codeword identify the position of the particular symbol as the binary value of (*m* - 1). The last 4 bits identify the total number of the symbols to be concatenated in the Structured Append format as the binary value of (17 - *n*). The 4-bit patterns shall conform with those defined in Table 8.

Table 8 — Structured Append symbol position bits

Symbol position	Bits 1234	Total number of symbols	Bits 5678
1	0000		
2	0001	2	1111
3	0010	3	1110
4	0011	4	1101
5	0100	5	1100
6	0101	6	1011
7	0110	7	1010
8	0111	8	1001
9	1000	9	1000
10	1001	10	0111
11	1010	11	0110
12	1011	12	0101
13	1100	13	0100
14	1101	14	0011
15	1110	15	0010
16	1111	16	0001

EXAMPLE To indicate the 3rd symbol of a set of 7, this shall be encoded thus:

3rd position: 0010

Total 7 symbols: 1010

Bit pattern: 00101010

Codeword value: 42

5.6.3 File identification

The file identification is defined by the value of its two codewords. Each file identification codeword may have a value 1 to 254, allowing 64516 different file identifications. The purpose of the file identification is to increase the probability that only logically linked symbols are processed as part of the same message.

5.6.4 FNC1 and Structured Append

If Structured Append is used in conjunction with FNC1 (see 5.2.4.6), the first four codewords shall be used for Structured Append and the fifth and sixth codewords are available for FNC1 usage. FNC1 shall not be repeated in these positions in the second and subsequent symbols, except when used as a field separator.

5.6.5 Buffered and unbuffered operation

The message within a Structured Append sequence can be buffered in the reader in its entirety and transmitted after all of the symbols have been read. Alternatively, the reader may transmit the decoded data in each symbol as it is read. In this unbuffered operation, the ECI protocol for structured append (specified in AIM ITS 04/001, Part 1) defines a control block that shall be prefixed to the beginning of the data transmitted for each symbol.

5.7 Error detection and correction

5.7.1 Reed-Solomon error correction

ECC 200 symbols employ Reed-Solomon error correction. For ECC 200 symbols with less than 255 total codewords, the error correction codewords are calculated from data codewords with no interleaving. For ECC 200 symbols with more than 255 total codewords, the error correction codewords are calculated from data codewords with the interleaving procedure described in Annex A. Each ECC 200 symbol has a specific number of data and error correction codewords which are divided into a specific number of blocks, as defined in Table 7, and to which the interleaving procedure defined in Annex A is applied.

The polynomial arithmetic for ECC 200 shall be calculated using bit-wise modulo 2 arithmetic and byte-wise modulo 100101101 (decimal 301) arithmetic. This is a Galois field of 2^8 with 100101101 representing the field's prime modulus polynomial: $x^8 + x^5 + x^3 + x^2 + 1$. Sixteen different generator polynomials are used for generating the appropriate error correction codewords. These are given in E.1.

5.7.2 Generating the error correction codewords

The error correction codewords are the remainder after dividing the data codewords by a polynomial *g(x)* used for Reed-Solomon codes (see E.1).

NOTE If this calculation is performed by "long division" the symbol data polynomial must first be multiplied by x^k.

The data codewords are the coefficients of the terms of a polynomial with the coefficient of the highest term being the first data codeword and the lowest power term being the last data codeword before the first error correction codeword. The highest order coefficient of the remainder is the first error correction codeword and the zero power coefficient is the last error correction codeword and the last codeword. This can be implemented by using the division circuit as shown in Figure 5. The registers b_0 through b_{k-1} are initialised as zeros. There are two phases to generate the encoding. In the first phase, with the switch in the down position

the data codewords are passed both to the output and the circuit. The first phase is complete after *n* clock pulses. In the second phase (*n* + 1 ... *n* + *k* clock pulses), with the switch in the up position, the error correction codewords ε_{k-1}, ... ,ε_0 are generated by flushing the registers in order while keeping the data input at 0. The codewords output from the shift register are in the order that they are to be placed in the symbol. If interleaving is used, the codewords will not be placed in consecutive symbol characters. (See Annex A).

Note: n and k are defined in 3.2 as the number of data codewords and the number of error correction codewords respectively.

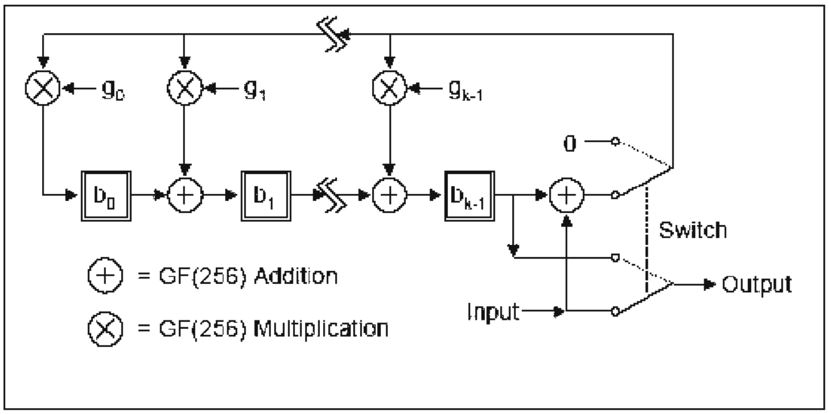


Figure 5 — Error correction codeword encoding circuit

5.7.3 Error correction capacity

The error correction codewords can correct two types of erroneous codewords: erasures (erroneous codewords at known locations) and errors (erroneous codewords at unknown locations). An erasure is an unscanned or undecodable symbol character. An error is a misdecoded symbol character. The number of erasures and errors that can be corrected is given by the following formula:

$e + 2t \leq d - p$

where:

e = number of erasures

t = number of errors

d = number of error correction codewords

p = number of codewords reserved for error detection.

In the general case, *p* = 0. However, if most of the error correction capacity is used to correct erasures, then the possibility of an undetected error is increased. Whenever the number of erasures is more than half the number of error correction codewords, *p* = 3. For small symbols (10 x 10, 12 x 12, 8 x 18, and 8 x 32), erasure correction should not be used (*e* = 0 and *p* = 1).

5.8 Symbol construction

Given the codeword sequence obtained in the previous sections, an ECC 200 symbol is constructed using the following steps:

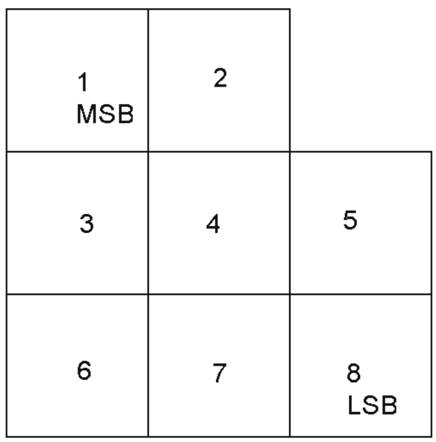
a) Place codeword modules in a mapping matrix

b) Insert alignment pattern modules, if any

c) Place finder modules along the perimeter

5.8.1 Symbol character placement

Each symbol character shall be represented by eight modules which are nominally square in shape; each module represents a binary bit. A dark module is a one and a light module is a zero. The eight modules are in order from left to right and top to bottom to form a symbol character as shown in Figure 6. Because the symbol character shape defined in Figure 6 cannot be perfectly nested at the symbol boundary, some symbol characters are split into portions. Symbol character placement is defined in the C language program in F.1, described in F.2 and illustrated in F.3.



LSB = Least significant bit
MSB = Most significant bit

Figure 6 — Representation of a codeword in a symbol character for ECC 200

5.8.2 Alignment Pattern module placement

This step is only needed for larger matrices: square: 32 x 32 and larger rectangular: 8 x 32, 12 x 36 and larger The mapping matrix is sub-divided into data regions, of the sizes defined in Table 7, for the chosen symbol format. The data regions are separated from each other by two-module-wide alignment patterns. This will result in some of the symbol characters being split between two adjacent data regions. For square matrices, the alignment patterns are placed between the data regions horizontally and vertically in pairs with a total alignment pattern count of 2, 6, or 10 as shown in Figures D.1 - D.3. For rectangular matrices, only a single vertical alignment pattern is placed between the data regions as shown in Figure D.4.

5.8.3 Finder Pattern module placement

Modules are placed along the perimeter of the matrix to construct the finder pattern as described in Section 4.3.1.

6 ECC 000 - 140 requirements

6.1 Use recommendations

For new applications or open systems ECC 200 is recommended (See Clause 5). There is no known application where ECC 200 will be more likely to succumb to symbol damage than ECC 000 to 140 for a given symbol size.

6.2 Encode procedure overview

This section provides an overview of the encoding procedure. Following sections will provide more details. An example encode for ECC 050 is given in Annex Q. The following steps convert user data to an ECC 000 - 140 symbol:

Step 1: Data encodation

> The user data is analysed to identify the variety of different characters to be encoded. For maximum compaction efficiency, the lowest level encodation scheme capable of encoding the data should be selected. If the user does not specify the matrix size, then choose the smallest size that accommodates the data. The result of this step is called the Encoded Data Bit Stream.

Step 2: Data prefix construction

> A Data Prefix Bit Stream is constructed from the Format ID, CRC, and Data Length bit fields. This Data Prefix Bit Stream is prefixed to the Encoded Data Bit Stream to produce the Unprotected Bit Stream.

Step 3: Error checking and correction

> The Unprotected Bit Stream is processed by the user specified convolutional coding encode algorithm to produce the Protected Bit Stream. This step is omitted for ECC 000.

Step 4: Header and trailer construction

> A header containing only the ECC bit field is prefixed to the Protected Bit Stream. A trailer containing pad bits (zeros) is appended to the Protected Bit Stream. The Protected Bit Stream with the header and trailer added is called the Unrandomised Bit Stream.

Step 5: Pattern randomising

> The Unrandomised Bit Stream is processed by the pattern randomising algorithm and produces the Randomised Bit Stream.

Step 6: Module placement in matrix

> Modules are placed in a matrix to construct the finder pattern. The Randomised Bit Stream is placed into the matrix one module at a time according to the data module placement algorithm given in Annex H. Figure 7 shows the various bit streams during the encode process.

6.3 Data encodation

The data shall be encoded using one of six encodation schemes (see Table 9). The encodation scheme is fixed for the entire symbol, and thus the selection of the most appropriate encodation scheme can have a considerable effect on the number of bits required to encode any given data. The same data may be represented in ECC 000 - 140 symbols in different ways through the use of the different encodation schemes. The character sets of all the encodation schemes, except the 8-bit byte scheme, are given in Annex I. The 8-bit byte scheme is user definable. The most efficient scheme to use is the lowest base number scheme

which is capable of encoding all the characters in the message. Thus if all the characters can be encoded in Base 27, it is not efficient to use Base 37, Base 41 or ASCII.

Table 9 — Encodation schemes

Encodation scheme	Characters	Bits per data character
Base 11	Numeric data	3,5
Base 27	Upper-case alphabetic	4,8
Base 37	Upper-case alphanumeric	5,25
Base 41	Upper-case alphanumeric and punctuation	5,5
ASCII	Full 128 ASCII set	7
8-bit Byte	User defined	8

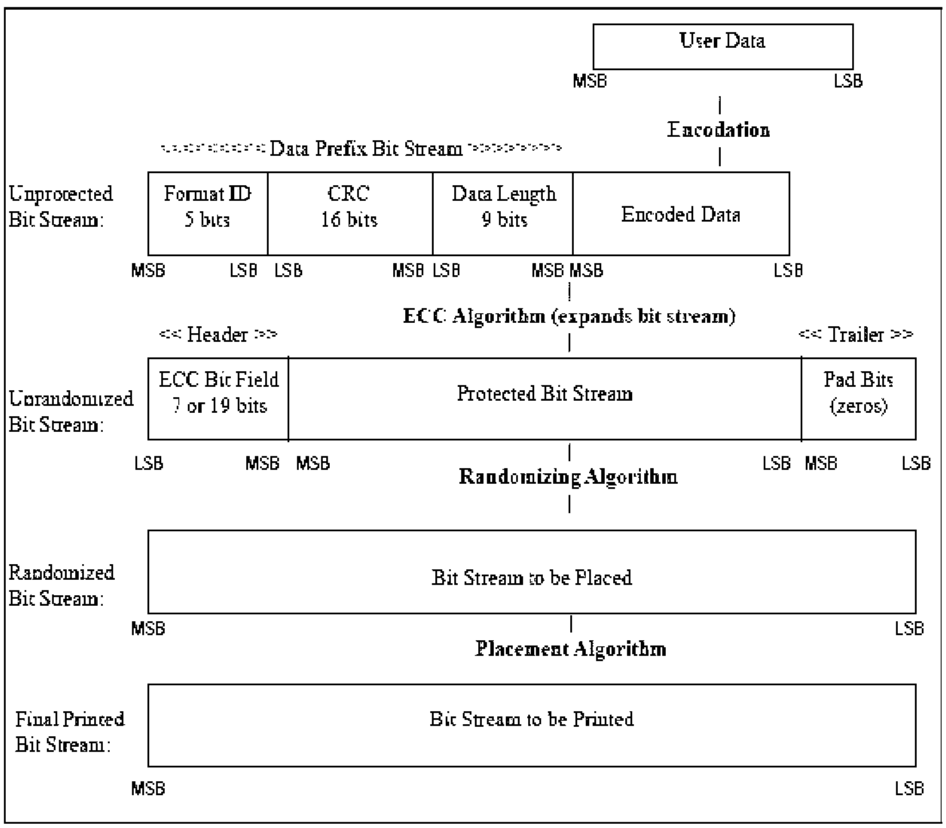


Figure 7 — ECC 000-140 encode process bit streams

To determine the appropriate encodation scheme, the data to be encoded should be analysed. The character sets of each of the Base *N* encodation schemes should be compared with the data character set to be encoded starting with the Base 11 character set. If this is suitable then it should be used, if not, the comparisons should continue with Base 27, Base 37 and Base 41, until the appropriate lowest level encodation scheme is found. If data characters beyond the capability of Base 41 need to be encoded, the ASCII set should be used, unless characters are beyond this; in which case the 8-bit byte set should be used.

For all encodation schemes, each compressed sequence of 4 to 24 bits is placed into the Encoded Bit Stream in reverse order (LSB first). This means that each individual compressed sequence is composed, then reversed, and output immediately to the Encoded Bit Stream. This does not mean that a complete compressed bit stream is formed, then reversed.

The details of each encodation scheme are given in the following clauses.

6.3.1 Base 11 - Numeric encodation

The Base 11 (Numeric) encodation scheme encodes 6 data characters as 21 bits, achieving an encodation density of 3,5 bits per data character. The Base 11 code set enables the following 11 characters to be encoded:

0 to 9

space

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 11 code values, as given in Annex I. In the second phase, the Base 11 code values shall be compacted using a Base 11 to Base 2 conversion according to the procedures defined in I.1.

6.3.2 Base 27 - Upper-case Alphabetic encodation

The Base 27 (Upper-case Alphabetic) encodation scheme encodes 5 data characters as 24 bits, achieving an encodation density of 4,8 bits per data character. The Base 27 code set enables the following 27 characters to be encoded:

A to Z

space

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 27 code values, as given in Annex I. In the second stage, the Base 27 code values shall be compacted using a Base 27 to Base 2 conversion according to the procedures defined in I.2.

6.3.3 Base 37 - Upper-case Alphanumeric encodation

The Base 37 (Upper-case Alphanumeric) encodation scheme encodes 4 data characters as 21 bits, achieving an encodation density of 5,25 bits per data character. The Base 37 code set enables the following 37 characters to be encoded:

A to Z

0 to 9

space

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 37 code values, as given in Annex I. In the second stage, the Base 37 code values shall be compacted using a Base 37 to Base 2 conversion according to the procedures defined in I.3.

6.3.4 Base 41 - Upper-case Alphanumeric plus Punctuation encodation

The Base 41 (Upper-case Alphanumeric plus Punctuation) encodation scheme encodes 4 data characters as 22 bits, achieving an encodation density of 5,5 bits per data character. The Base 41 code set enables the following 41 characters to be encoded:

A to Z

0 to 9

space

. (period)

, (comma)

- (minus or hyphen)

/ (forward slash or solidus)

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 41 code values, as given in Annex I. In the second stage, the Base 41 code values shall be compacted using a Base 41 to Base 2 conversion according to the procedures defined in I.4.

6.3.5 ASCII encodation

The ASCII encodation scheme enables all 128 characters from ISO/IEC 646 to be encoded. Each data character shall be encoded as a 7-bit byte equivalent to the decimal value shown in the ASCII column of Table I.1 of Annex I.

6.3.6 8-bit byte encodation

The 8-bit byte encodation scheme shall be used for closed applications, where the data interpretation shall be determined by the user. Each data character shall be encoded as an 8-bit byte.

6.4 User selection of error correction level

6.4.1 Selection of error correction level

ECC 000 - 140 symbols offer five levels of error correction using convolutional code error correction, as set out in Table 10. In an application, it is important to understand that these error correction levels result in the generation of a proportional increase in the number of bits in the message (and hence increase in the size of the symbol), and offer different levels of error recovery.

Table 10 — Error correction, error recovery and overhead percentages

Error correction code level	Maximum % damage	% increase in user bits from ECC 000
000	none	none
050	2,8	33
080	5,5	50
100	12,6	100
140	25	300

6.4.2 Other error correction levels based on convolutional code algorithms

Other levels of error correction, based on convolutional code algorithms, have been used in Data Matrix applications implemented prior to the publication of this International Standard. Information on these non-standard Error Correction levels is available from AIM Inc. Such symbols do not conform with this International Standard.

6.5 Constructing the Unprotected Bit Stream

Figure 7 illustrates that the Unprotected Bit Stream has the Data Prefix Bit Stream as a prefix to the encoded data bits. The component parts of the Data Prefix Bit Stream are defined below.

6.5.1 Format ID Bit Field

The format ID defines the data encodation scheme. The format ID has a decimal value for the purposes of definition and a 5-bit segment value for encoding as defined in Table 11.

Table 11 — Encoding the Format ID

Format ID	Encodation scheme	Binary segment value
		MSB LSB
1	Base 11	00000
2	Base 27	00001
3	Base 41	00010
4	Base 37	00011
5	ASCII	00100
6	8-bit Byte	00101

6.5.2 CRC Bit Field

The CRC Bit Field is generated by the CRC algorithm. The CRC Value is generated from the original user data as 8-bit bytes before encodation and so produces an independent error check on the user data. Annex J describes the complete procedure for generating the CRC Value.

6.5.3 Data Length Bit Field

The Data Length Bit Field is 9 bits in length and represents, as a binary value, the number of user data characters being encoded.

6.5.4 Data prefix construction

The Data Prefix Bit Stream is constructed as 30 bits as illustrated in Figure 8.

Format ID 5 bits	CRC Value 16 bits	Data Length 9 bits
MSB LSB	LSB MSB	LSB MSB

Figure 8 — Structure of Data Prefix Bit Stream

NOTE Some bit fields start with the MSB, others start with the LSB.

6.5.5 Completing the Unprotected Bit Stream

The encoded data bits are added as a suffix to the Data Prefix Bit Stream to construct the Unprotected Bit Stream.

6.6 Constructing the Unrandomised Bit Stream

Figure 7 illustrates that the Unrandomised Bit Stream has three constituent parts:

a) Header

b) Protected Bit Stream

c) Trailer

The component parts shall be generated as defined below.

6.6.1 Header construction

The header of the Unrandomised Bit Stream contains the ECC Bit Field, which identifies the convolutional code structure used to protect the data encoded in the symbol. The ECC Bit Field is 7 or 19 bits long and the values are shown in Table 12.

Table 12 — ECC Bit Field

ECC Level	**Binary Segment Identifier** **MSB LSB**
000	1111110
050	0001110000000001110
080	1110001110000001110
100	1111111110000001110
140	1111110001110001110

6.6.2 Applying convolutional coding to create the Protected Bit Stream

One of the five error correction levels shall be applied. The selection criteria are defined in Section 6.4. No error correction is applied for ECC 000, so the Unprotected Bit Stream becomes the Protected Bit Stream. For the other four error correction levels, convolutional coding is applied. This expands the user data proportionally throughout its length. The encoded bit stream shall be created by processing the unprotected bit stream through the appropriate error correction state machine and reading the results. The circuit diagrams of the four state machines for ECC 050 to 140 are given in Annex K.

6.6.3 Trailer construction

A Trailer containing pad bits (zeros) is appended to the Protected Bit Stream. Pad bits shall be added at the end of the bit stream to ensure that the square root of the total number of bits in the Unrandomised Bit Stream shall be an odd integer between 7 and 47. This ensures that the symbol is square.

6.6.4 Completing the Unrandomised Bit Stream

The Protected Bit Stream, with the header and trailer added, is called the Unrandomised Bit Stream and is shown in Figure 7.

6.7 Pattern randomising

The Unrandomised Bit Stream is processed by the pattern randomising algorithm and produces the Randomised Bit Stream. The pattern randomising algorithm consists of a bitwise XOR operation between the Unrandomised Bit Stream and the Master Random Bit Stream as given in Annex L starting with the MSB position and continuing for the length of the Unrandomised Bit Stream.

6.8 Module placement in matrix

The size of the sides of the data module grid is given by the odd integer square root (between 7 and 47) calculated in the procedure defined in 6.6.3. The Randomised Bit Stream is placed into the matrix one module at a time according to the data module placement grids given in Annex H. The finder pattern (as defined in 4.3.1) shall be placed to produce an external border to the data module grid.

7 Symbol dimensions

7.1 Dimensions

Data Matrix symbols shall conform to the following dimensions:

X dimension: the width of a module shall be specified by the application, taking into account the scanning technology to be used, and the technology to produce the symbol.

finder pattern: the width of the finder pattern shall equal *X*.

alignment pattern: the width of the alignment pattern shall equal 2*X*.

Quiet zone: The minimum quiet zone is equal to *X* on all four sides. For applications with moderate to excessive reflected noise in close proximity to the symbol, a Quiet Zone of 2*X* to 4*X* is recommended

8 Symbol quality

Data Matrix symbols shall be assessed for quality using the 2D matrix bar code symbol print quality guidelines defined in ISO/IEC 15415, as augmented and modified below.

Some marking technologies may not be able to produce symbols conforming to this specification without taking special precautions. Annex T gives additional guidance to help any printing system achieve valid Data Matrix symbols.

8.1 Symbol quality parameters

8.1.1 Fixed pattern damage

Annex M defines the measurement and grading basis for Fixed Pattern Damage.

NOTE As provided for in Annex A of ISO/IEC 15415, the measurements and values defined in Annex M of this International Standard override those indicated in Annex A of ISO/IEC 15415.

8.1.2 Scan grade and overall symbol grade

The scan grade shall be the lowest of the grades for symbol contrast, modulation, fixed pattern damage, decode, axial non-uniformity, grid non-uniformity and unused error correction in an individual image of the symbol. The overall symbol grade is the arithmetic mean of the individual scan grades for a number of tested images of the symbol.

8.1.3 Grid non-uniformity

The ideal grid is calculated by using the four corner points of the sampling grid for each data region and subdividing it equally in both axes.

8.1.4 Decode

The reference decode algorithm specified in this international standard shall be applied to determine the grade for Decode. A failure of the reference decode algorithm to successfully decode the symbol shall result in a grade of 0 for decode.

8.2 Process control measurements

A variety of tools and methods can be used to perform useful measurements for monitoring and controlling the process of creating Data Matrix symbols. These are described in Annex R. These techniques do not constitute a print quality check of the produced symbols (the method specified earlier in this clause and Annex M is the required method for assessing symbol print quality) but they individually and collectively yield good indications of whether the symbol print process is creating workable symbols.

AC2 9 Reference decode algorithm for Data Matrix

This reference decode algorithm finds a Data Matrix symbol in an image and decodes it.

a) Define measurement parameters and form a digitised image:

1) Define a distance d_{min} which is 7,5 times the aperture diameter defined by the application. This will be the minimum length of the "L" pattern's side.

2) Define a distance g_{max} which is 7,5 times the aperture diameter. This is the largest gap in the "L" finder that will be tolerated by the finder algorithm in step b).

3) Define a distance m_{min} which is 1,25 times the aperture diameter. This would be the nominal minimum module size when the aperture size is 80% of the symbol's X dimension.

4) Form a black/white image using a threshold determined according to the method defined in ISO/IEC 15415. AC2

AC2 b) Search horizontal and vertical scan lines for the two outside edges of the Data Matrix "L":

1) Extend a scan line horizontally in both directions from the centre point of the image. Sample along the scan line. For each white/black or black/white transition found along the scan line resolved to the pixel boundary:

   i) Follow the edge upward sampling pixel by pixel until either it reaches a point 3,5m_{min} distant from the intersection of the scan line and the edge starting point, or the edge turns back toward the intersection of the scan line and the edge - the starting point.

   ii) Follow the edge downward pixel by pixel until either it reaches a point 3,5m_{min} distant from the intersection of the scan line and the edge starting point, or the edge turns back toward the intersection of the scan line and the edge - the starting point.

   iii) If the upward edge reaches a point 3,5m_{min} from the starting point:

      I) Plot a line A connecting the end points of the upward edge.

      II) Test whether the intermediate edge points lie within 0,5m_{min} from line A. If so, continue to step III. Otherwise proceed to step 1)iv) to follow the edge in the opposite direction.

      III) Continue following the edge upward until the edge departs 0,5m_{min} from line A. Back up to the closest edge point greater than or equal to m_{min} from the last edge point along the edge before the departing point and save this as the edge end point. This edge point should be along the "L" candidate outside edge.

      IV) Continue following the edge downward until the edge departs 0,5m_{min} from line A. Back up to the closest edge point greater than or equal to m_{min} from the last edge point along the edge before the departing point and save this as the edge end point. This edge point should be along the "L" candidate outside edge.

      V) Calculate a new adjusted line A1 that is a "best fit" line to the edge in the two previous steps. The "best fit" line uses the linear regression algorithm (using the end points to select the proper dependent axis, i.e. if closer to horizontal, the dependent axis is x) applied to each point. The "best fit" line terminates lines at points p1 and p2 that are the points on the "best fit" line closest to the endpoints of the edge.

      VI) Save the line A1 segment two end points, p1 and p2. Also save the colour of the left side of the edge viewed from p1 to p2.

   iv) If step iii) failed or did not extend upward by 3,5m_{min} in step iii)IV), test if the downward edge reaches a point 3,5m_{min} from the starting point. If so, repeat the steps in iii) but with the downward edge.

   v) If neither steps iii) or iv) were successful, test if both the upward and downward edges terminated at least 2m_{min} from the starting point. If so, form an edge comprised of the appended 2m_{min} length upward and downward edge segments and repeat the steps in iii) but with the appended edge.

   vi) Proceed to and process the next transitions on the scan line, repeating from step i), until the boundary of the image is reached.

2) Extend a scan line vertically in both directions from the centre point of the image. Look for line segments using the same logic in step 1) above but following each edge transition first left and then right. AC2

AC2 3) Search among the saved line A1 segments for pairs of line segments that meet the following four criteria:

i) If the two lines have the same p1 to p2 directions, verify that the closer of the interline p1 to p2 distances is less than g_{max}. If the two lines have opposite p1 to p2 directions, verify that the closer of the interline p1 to p1 or p2 to p2 distances is less than g_{max}.

ii) Verify that the two lines are co-linear within 5 degrees.

iii) Verify that the two lines have the same saved colour if their p1 to p2 directions are the same or that the saved colours are opposite if their p1 to p2 directions are opposite to each other.

iv) Form two temporary lines by extending each line to reach the point on the extension that is closest to the furthest end point of the other line segment. Verify that the two extended lines are separated by less than 0,5m_{min} at any point between the two extended lines.

4) For each pair of lines meeting the criteria of step 3) above, replace the pair of line segments with a longer A1 line segment that is a "best fit" line to the four end points of the pair of shorter line segments. Also save the colour of the left side of the edge of the new longer line viewed from its p1 endpoint to its p2 endpoint.

5) Repeat steps 3) and 4) until no more A1 line pairs can be combined.

6) Select line segments that are at least as long as d_{min}. Flag them as "L" side candidates.

7) Look for pairs of "L" side candidates that meet the following three criteria:

i) Verify that the closest points on each line are separated by less than 1,5g_{max}.

ii) Verify that they are perpendicular within 5 degrees.

iii) Verify that the same saved colour is on the inside of the "L" formed by the two lines. Note that if one or both lines extend past their intersection, then the two or four "L" patterns formed will need to be tested for matching colour and maintaining a minimum length of d_{min} for the truncated side or sides before they can become "L" candidates.

8) For each candidate "L" pair found in step 7) form an "L" candidate by extending the segments to their intersection point.

9) If the "L" candidate was formed from line segments with the colour white on the inside of the "L", form a colour inverted image to decode. Attempt to decode the symbol starting with the appropriate normal or inverted image starting from step d) below using each of the "L" candidates from step 8) as the "L" shaped finder. If none decode, proceed to step c).

c) Maintain the line A1 line segments and "L" side candidates from the previous steps. Continue searching for "L" candidates using horizontal and vertical scan lines offset from previous scan lines:

1) Using a new horizontal scan line 3m_{min} above the centre horizontal scan line, repeat the process in step b)1), except starting from the offset from the centre point, and then b)3) through b)9). If there is no decode, proceed to the next step.

2) Using a new vertical scan line 3m_{min} left of the centre vertical scan line, repeat the process in step b)2), except starting from the offset from the centre point, and then steps b)3) through b)9). If there is no decode, proceed to the next step.

3) Repeat step 1) above except using a new horizontal scan line 3m_{min} below the centre horizontal scan line. If there is no decode, repeat step 2) above except using a new vertical scan line 3m_{min} right of the centre vertical scan line. If there is no decode, proceed to step 4) below. AC2

AC2 4) Continue processing horizontal and vertical scan lines as in steps 1) through 3) that are $3m_{min}$ above, then left, then below, then right of the previously processed scan lines until either a symbol is decoded or the boundary of the image is reached.

d) First assume that the candidate area contains a square symbol. If the area fails to decode as a square symbol, then try to find and decode a rectangular symbol starting from procedure j). For a square symbol, first plot a normalised graph of transitions for the equal sides of the candidate area in order to find the alternating module finder pattern:

1) Project a line through the candidate area bisecting the interior angle of the two sides of the "L" found above as shown in figure 9. Define the two equal areas formed by the bisecting line as the right side and the left side as viewed from the corner of the "L".

2) For each side, form a line called a “search line” between a point d_{min} distance from the corner along the “L” line, parallel to the other “L” side line, and extending to the bisecting line as shown in Figure 9.

3) Move each search line away from the corner of the “L” as shown in Figure 9, lengthening each line as it expands to span its two bounding lines, the “L” line and the bisecting line. Keep each search line parallel to the other “L” side line. As each side is moved by the size of an image pixel, count the number of black/white and white/black transitions, beginning and ending the count with transitions from the colour of the “L” side to the opposite colour. A transition from one colour to the other is to be counted only when the current search line as well as the search lines immediately above and below have the same colour, opposite to the previously counted transition colour. Plot the number of transitions multiplied by the length of the longest “L” side divided by the current length of the search line measured between the two bounding lines:

T = (number of transitions) (“L” max. line length) / (search line length).

This formula normalises T to keep it from increasing because the line lengthens.

Continue to calculate the T values until the search line is longer than the longest axis of the candidate area plus 50%.

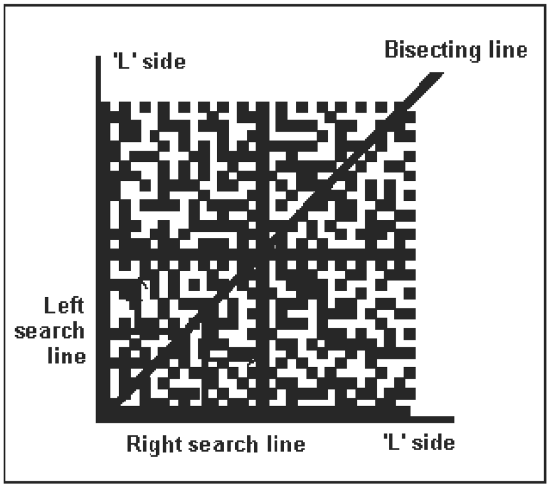


Figure 9 — Expanding search lines AC2

AC2 4) Form a plot of the *T* values for each side, where the *Y*-axis is the *T* value and the *X*-axis is the search line's distance from the corner of the "L". A sample plot is shown in Figure 10.

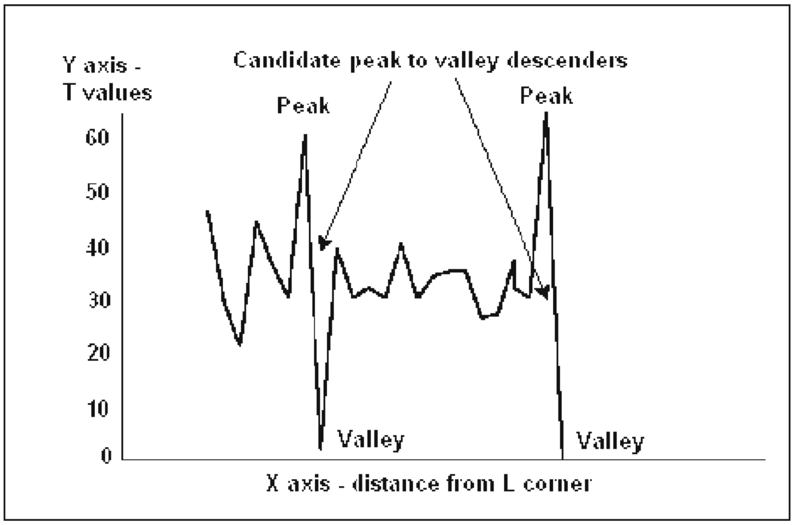


Figure 10 — Example plot of T as the search line expands

5) Starting from the *T* value with the smallest *X* in the right side's plot and then increasing *X*, find the first instance of a T_s value (T_s = maximum of zero and *T* - 1) that is less than 15% of the preceding local maximum *T* value, provided that *T* value is greater than 1. Increment this *X* value until the number of transitions stops decreasing. If the number of transitions does not increase, increment the *X* value once more. Refer to this *X* value as the valley. Increment the local maximum's *X* value until the number of transitions decreases and refer to this *X* as the peak. Refer to the average of the peak and valley *X* values as the descending line *X* value. The search line at the peak may correspond to an alternating finder pattern side. At the valley, the search line may correspond to the solid dark interior line or a light quiet zone.

6) Find the peak and valley in the left side's plot whose descending line *X* value most closely matches the right peak and valley's descending line *X* value. If returning to this step from a later step, consider additional left peaks and valleys, ordered in terms of how closely they match the right peak and valley. However, any left peak and valley under consideration must be checked to ensure that the absolute difference between the right and left peak *X* values is less than 15% of the average of the two peak *X* values and that the absolute difference between the right and left valley *X* values is less than 15% of the average of the two valley *X* values. The 15% specifies the maximum allowed foreshortening.

7) The right side's valley search line, the left side's valley search line, and the two sides of the "L" outline a possible symbol's data region. Process the data region according to step e). If the decode fails, find the next left peak and valley from step d)6). Once all left peaks and valleys have been discarded, discard the right side peak and valley and continue searching from step d)5) for the next right peak and valley.

e) For each of the two sides of the alternating pattern, find the line passing through the centre of the alternating light and dark modules:

1) For each side, form a rectangular region bounded by the side's peak and valley search lines as the longer two sides of the rectangle, and the "L" side and the other side's valley search line as the shorter two sides, as shown in Figure 11. AC2

AC2

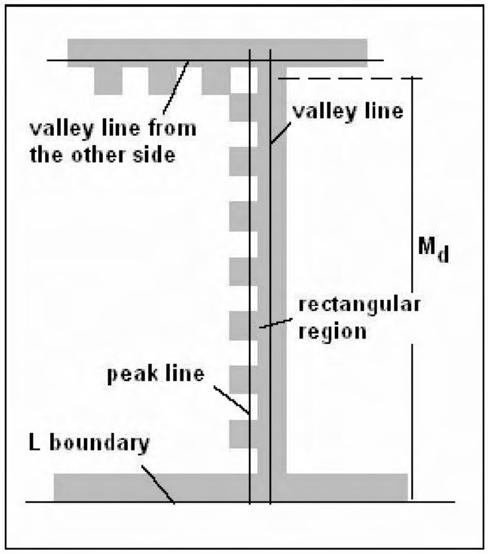


Figure 11 — Rectangular region construction

2) Within the rectangular region, find pixel edge pairs on the outside boundary of teeth:

i) Traverse test lines starting with and parallel to the valley line looking for transitions to the opposite colour normally orthogonal to the test line. Select only transitions that are either dark to light or light to dark where the first colour matches the predominate colour of the image along the valley line.

ii) If the number of transitions found is less than 15% of the number of pixels comprising the valley line, and the test line is not the peak line, move the test line toward the peak line by nominally one pixel and repeat step i), now considering new transitions in addition to those already found. If the 15% criterion is met or the peak line is reached, continue to the next step, otherwise continue searching from step d)6) for the next left peak and valley.

iii) Calculate a preliminary "best fit line" with linear regression using the points on the edge between the selected pixel pairs.

iv) Discard the 25% of the points which are furthest from the preliminary "best fit line". Calculate a final "best fit line" with linear regression using the remaining 75% of points. This line should pass along the outside of the alternating pattern, shown as the "best fit line" in Figure 12.

3) For each side, construct a line parallel to the step e)2) line which is offset toward the “L” corner by the perpendicular distance from the “L” corner to the peak search line divided by twice the number of transitions in the peak search line plus one:

Offset = distance to the peak line / ((number of transitions + 1) * 2)

Each of the two constructed lines should correspond to the mid-line of the alternating module pattern on that side, see Figure 12. AC2

AC2

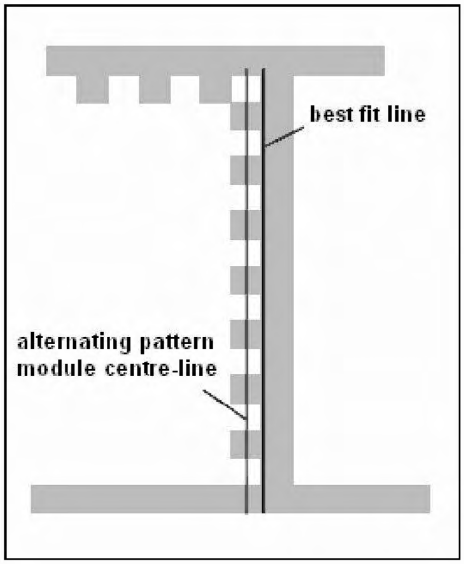


Figure 12 — Alternating pattern module centre-line

f) For each side, measure the edge-to-edge distances in the alternating pattern:

1) Bound the alternating pattern mid-line constructed in step e)3) by the adjacent "L" line and the other alternating pattern mid-line from step e)3). Call the length of this line M_d (see Figure 11).

2) Along the bounded mid-line, measure the edge-to-edge distances between all the similar edges of all two-element pairs, i.e. dark/light and light/dark element pairs. Begin and end the edge-to-edge measurements with edges transitioning from the "L" colour to the opposite colour.

3) Select the median edge-to-edge measurement and set the current edge-to-edge measurement estimate, *EE_Dist*, to the median measurement.

4) Discard all element pairs with edge-to-edge measurements that differ more than 25% from *EE_Dist*.

g) For each side, find the centre points of the alternating pattern modules:

1) Using the remaining element pair measurements from f)4), calculate the average ink spread (vertical or horizontal depending on the segment side) by the average of the element pair's ink spread, where *bar* is the dark element width and *space* is the light element width in a remaining element pair:

$$ink\_spread = \text{Average}\ (\ (bar - ((bar + space) / 2)) / ((bar + space) / 2)\ )$$

2) Calculate the centre of the bar in the median element pair using the following offset into the bar from the outside edge of the bar in the median pair:

$$\text{offset} = (EE\_Dist * (1 + ink\_spread)) / 4$$

If there is more than one median element pair, choose a single pair using the following process:

i) Order the edges (excluding the "L" finder edge) by their distance from the "L" finder edge. There are an odd number of these edges because the edges start and end on a dark to light transition going away from the "L" finder.

ii) Call the middle edge in the list the centre edge. AC2

AC2 iii) Calculate the (odd number of) element pair edge-to-edge distances and find their median *EE_Dist*.

iv) Select the one or more element pairs with length *EE_Dist*.

v) Among those pairs identify the one or two element edge pairs that has an edge closest to the centre edge.

vi) If there is still a tie, take the element pair that has the outer edge of the bar closest to centre edge.

vii) If there is still a tie, take the element pair that has an inner edge closest to the "L" finder.

3) Starting from the centre of the bar in the median element pair from step f)3) proceed in the direction of the space in the element pair until reaching the end of the bounded mid-line, calculate each element's centre, shown by the speckled pattern in Figure 13, by the following steps:

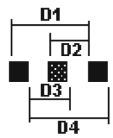


Figure 13 — Edge-to-edge measurements for finding an element centre

(While three bars and two spaces are shown in Figure 13, if a space is the element for which the centre is to be calculated, then the diagram would have three spaces instead of the bars and two bars instead of the spaces. For light elements adjacent to the element at the end of the mid-line, either *D1* or *D4* measurements are omitted as they would fall outside the symbol's or segment's measurable element boundaries.)

i) Calculate a point p1 along the mid-line which is *EE_Dist*/2 from the previously calculated element centre in the direction of the new element.

ii) Calculate d_1 through d_4 where:

$d_1 = D1 / 2$

$d_2 = D2$

$d_3 = D3$

$d_4 = D4 / 2$

iii) If one of the values d_1 through d_4 is within 25% of *EE_Dist*, select the one which is closest to *EE_Dist*, and set the new *EE_Dist* to be the average of the current *EE_Dist* and the selected d_1 through d_4 distance.

I) If d_1 or d_4 are selected, select the corresponding *D1* or *D4* edge closest to the element, the centre of which is to be calculated. Offset this edge by *(ink_spread/2) * (EE_Dist/2)* in the appropriate direction (i.e., if *ink_spread* is positive, the offset will move the edge toward the space included in the distance *D1* or *D4* and if negative, the offset will move away from this space). Calculate a point p2 along the mid-line which is 0,75 times the selected d_1 or d_4 value from the offset edge and toward the element centre to be calculated.

II) If d_2 or d_3 are selected, select the corresponding *D2* or *D3* edge closest to the element the centre of which is to be calculated. Offset this edge by *(ink_spread/2) * (EE_Dist/2)* in the appropriate direction (i.e., if *ink_spread* is positive, the offset will move the edge toward the space included in the distance *D2* or *D3* and if negative, the offset will move away from this space). Calculate a point p2 along the mid-line which is 0,25 times the selected d_2 or d_3 value from the offset edge and toward the element centre to be calculated. AC2

AC2 III) Set the element's centre as halfway between p1 and p2.

iv) Otherwise if none of the values d_1 through d_4 is within 25% of *EE_Dist*, leave *EE_Dist* at its current value, use p1 as the new element's centre, and proceed to the next element.

4) Starting from the bar in the median element pair, and proceeding in the opposite direction from step 3), until reaching the other end of the bounded mid-line, calculate each element's centre, following the procedures in step 3).

h) If the number of modules in each side do not correspond to a valid first region, continue searching from step d)6) for the next left peak and valley. Otherwise plot the data module sampling grid in the data region by extending the alternating pattern module centres:

1) Extend each side's step e)3) mid-line and the opposite side's "L" line to form the vanishing point of the two nearly parallel or parallel extended lines.

2) Extend rays from each vanishing point passing through the step g) module centres of the nearly perpendicular step e)3) line.

3) The intersection of the two sets of nearly perpendicular rays should correspond to the centres of the data modules in the data region, as shown in Figure 14.

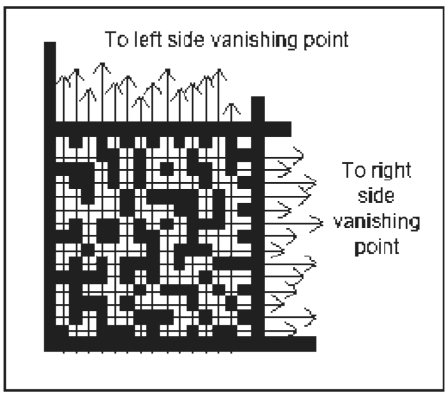


Figure 14 — Module sampling grid construction

i) Continue to fill in the remaining data regions:

1) When a data region is processed, form a new "L" for the next data section to the "left" or "above" using one of two processes:

i) If the new data region is still bounded on one side by the original "L" from procedure b), repeat from procedure c) to process the new data region using the selected set of points from step e)2) and the set of points on the "L" from step b)2) which lie beyond the step e)2) line.

ii) If the new data region is bounded on two sides by data regions, repeat from procedure c) to process the new data region using the selected set of points from step e)2) for each data region which are adjacent and bound the new region on two sides

2) If a data region does not match the number of modules in previously processed regions, trim the symbol to the largest number of regions which correspond to a legal symbol.

3) Decode the symbol with its one or more data regions starting with procedure k). AC2

AC2 4) If the current data region exhausts its last peak and valley, revert to the previous data region and continue searching from step d)6) for the next left peak and valley in that data region.

j) Find the data sections of a rectangular symbol.

1) For each side of the "L" move a line perpendicular to the side and scanning along the length of the other side of the "L". Keep each search line parallel to the other "L" side line. As each side is moved by the size of an image pixel, count the number of black/white and white/black transitions, beginning and ending the count with transitions from the colour of the "L" side to the opposite colour. A transition from one colour to the other is to be counted only when the current search line as well as the search lines immediately above and below have the same colour, opposite to the previously counted transition colour. As each side is moved by a pixel, plot the number of transitions, T. Continue until the parallel line moves further than the perpendicular leg of the "L" plus 10%.

2) Starting from the origin of the plot, for each direction, find the first instance of a T_s value (T_s = maximum of zero and T - 1) value that is less than 15% of the preceding local maximum T value, provided that T value is greater than 1. Increment this X value until the T value stops decreasing. If the T value does not increase, increment the X value once more. Refer to this X value as the valley. Increment the local maximum's X value until the T value decreases and refer to this X as the peak. Refer to the average of the peak and valley X Values as the descending line X value. The valley line at this point may form a side of a symbol or data region.

3) Find the alternating pattern lines for each side of the region similar to procedure e).

4) Plot the module sample grid in the data region or symbol as in procedures f), g), and h).

5) If the data region defined is not a valid rectangular symbol, try to form a new data region using further valid peak to valley plot transitions.

6) Process any additional regions as in procedure i).

7) If a valid data region or two regions are detected, attempt to decode the symbol as in procedures k) and l). If the region(s) were not valid or the decode fails, disregard the candidate area.

k) If the number of data modules is even or the symbol forms a valid rectangular symbol, decode the symbol using Reed-Solomon error correction:

1) Sample the data modules at their predicted centres. Black at the centre is a one and white is a zero.

2) Convert the eight module samples in the defined codeword patterns into 8-bit symbol character values.

3) Apply Reed-Solomon error correction to the symbol character values.

4) Decode the symbol characters into data characters according to the specified encodation schemes.

l) Otherwise the number of data modules is odd, so decode the symbol using convolution code error correction:

1) Sample the data modules at their predicted centres. Black at the centre is a one and white is a zero.

2) Apply the black/white balancing mask.

3) Use the bit ordering table to convert the data into a bit stream.

4) Apply the appropriate convolution code error correction.

5) Convert the bit stream to data characters according to the encodation scheme specified.

6) Verify that the CRC is correct. AC2

10 User guidelines

10.1 Human readable interpretation

Because Data Matrix symbols are capable of encoding thousands of characters, a human readable interpretation of the data characters may not be practical. As an alternative, descriptive text rather than the encoded text may accompany the symbol. The character size and font are not specified, and the message may be printed anywhere in the area surrounding the symbol. The human readable interpretation should not interfere with the symbol itself or the quiet zones.

10.2 Autodiscrimination capability

Data Matrix can be used in an autodiscrimination environment with a number of other symbologies. (See Annex S).

10.3 System considerations

Data Matrix applications must be viewed as a total system solution (see Annex T).

11 Transmitted data

This section describes the standard transmission protocol for compliant readers. These readers may be programmable to support other transmission options. All encoded data characters are included in the data transmission. The symbology control characters and error correction characters are not transmitted. More complex interpretations are addressed below.

11.1 Protocol for FNC1 (ECC 200 only)

When FNC1 appears in the first symbol character position (or in the fifth symbol character position of the first symbol of a Structured Append sequence), it shall signal that the data conforms to the GS1 Application Identifier standard format. FNC1 in any other later position in such symbols acts as a field separator. Transmission of symbology identifiers shall be enabled. The first FNC1 shall not be represented in the transmitted data, although its presence is indicated by the use of the appropriate option value (2) in the symbology identifier (see 11.5).

When used as a field separator, FNC1 shall be represented in the transmitted message by the ASCII character <$^{G}{}_{S}$> (ASCII value 29).

11.2 Protocol for FNC1 in the second position (ECC 200 only)

When FNC1 is in the second symbol character position (or in the sixth symbol character position of the first symbol of a Structured Append sequence), it shall signal that the data conforms to a particular industry standard format. Transmission of symbology identifiers shall be enabled. The first FNC1 shall not be represented in the transmitted data, although its presence is indicated by the use of the appropriate option value (3) in the symbology identifier (see 11.5).

The data encoded in the first symbol character shall be transmitted as normal at the beginning of the data. When used as a field separator, FNC1 shall be represented in the transmitted message by the ASCII character <$^{G}{}_{S}$> (ASCII value 29).

11.3 Protocol for Macro characters in the first position (ECC 200 only)

This protocol is used to encode two specific message headers and trailers in an abbreviated manner in ECC 200 symbols.

When a Macro character is in the first position a preamble and postamble shall be transmitted. If the first symbol character is 236 (i.e. encoding Macro 05), then the preamble [)>$^{R}{}_{S}$05$^{G}{}_{S}$ shall precede the encoded data that follows it. If the first symbol character is 237 (i.e. encoding Macro 06), then the preamble [)>$^{R}{}_{S}$06$^{G}{}_{S}$ shall precede the encoded data that follows it. The postamble $^{R}{}_{S}{}^{E}{}_{O_{T}}$ shall be transmitted after the data in both cases.

11.4 Protocol for ECIs (ECC 200 only)

In systems where ECIs are supported, the use of a symbology identifier prefix is required with every transmission. Whenever an ECI codeword is encountered, it shall be transmitted as the escape character 92_{DEC} (or $5C_{HEX}$), which represents the character "\" (backslash or reverse solidus) in the default interpretation. The next codeword(s) are converted into a 6-digit value, inverting the rules defined in Table 6. The 6-digit value is transmitted as the appropriate ASCII values (48 - 57). Application software recognising \nnnnnn should interpret all subsequent characters as being from the ECI defined by the 6-digit sequence. This interpretation remains in effect until the end of the encoded data or until another ECI sequence is encountered. If the backslash (byte 92_{DEC}) needs to be used as encoded data, transmission shall be as follows. Whenever (ASCII 92_{DEC}) occurs as data, two bytes of that value shall be transmitted, thus a single occurrence is always an escape character and a double occurrence indicates true data.

EXAMPLE

Encoded data: A\\B\C

Transmission: A\\\\B\\C

Use of the symbology identifier assures that the application can correctly interpret the escape character.

11.5 Symbology identifier

ISO/IEC 15424 provides a standard procedure for reporting the symbology which has been read, together with options set in the decoder and special features encountered in the symbol. Once the structure of the data (including the use of any ECI) has been identified, the appropriate symbology identifier should be added by the decoder as a preamble to the transmitted data. The symbology identifier is required if ECIs appear anywhere in the symbol, or if FNC1 is used as defined in 11.1 or 11.2. See Annex N for the symbology identifier and option values which apply to Data Matrix.

11.6 Transmitted data example

In this example, the two-character message "¶Ж" is to be encoded in ECC 200, using the ASCII encodation scheme. "¶" is represented by a byte value of 182 in Data Matrix's default character set (ECI 000003, which is equivalent to ISO 8859-1). "Ж" is a Cyrillic character not available in ECI 000003, but which can be represented in ISO 8859-5 (ECI 000007) by the same byte value of 182. The complete message can therefore be represented by inserting a switch to ECI 000007 after the first character, as follows: The symbol encodes the message <¶> <Switch to ECI 000007> <Ж>, using the following series of Data Matrix codewords: [Upper Shift] [55] [ECI] [8] [Upper Shift] [55], with decimal values of [235], [55], [241], [8], [235], [55].

NOTE 1 An Upper Shift character, followed by a codeword of value 55, encodes a byte value of 182.

NOTE 2 ECIs are encoded in Data Matrix as the ECI number plus one.

The decoder transmits the following bytes (including the symbology identifier prefix with an option value of 4, which indicates use of the ECI protocol):

93, 100, 52, 182, 92, 48, 48, 48, 48, 48, 55, 182

which, if viewed entirely in the default interpretation, would appear graphically as:]d4¶\000007¶

The decoder is responsible for signalling the switch to ECI 000007, but not for interpreting the result. ECI-aware software in the receiving application would delete the ECI escape sequence \000007, and the Cyrillic character "Ж" would be represented in a system-dependent manner (e.g., by changing the font in a desktop-publishing file). The final result would match the original message of "¶Ж".

Annex A
(normative)

ECC 200 interleaving process

A.1 Schematic illustration

Using the example of the 72 x 72 symbol size, four levels of interleaving are required to encode a total of 368 data codewords and 144 error correction codewords. These are divided into four blocks of 92 data codewords and 36 error correction codewords, a total block length of 128 codewords.

CODEWORD STREAM	data codewords d: 1, 2, 3, 4, …, …, 365, 366, 367, 368	error correction codewords ε: 1, 2, 3, 4, …, …, 141, 142, 143, 144
BLOCK 1	data codewords d: 1, 5, …, …, …, 361, 365	error correction codewords ε: 1, 5, …, …, 137, 141
BLOCK 2	data codewords d: 2, 6, …, …, …, 362, 366	error correction codewords ε: 2, 6, …, …, 138, 142
BLOCK 3	data codewords d: 3, 7, …, …, …, 363, 367	error correction codewords ε: 3, 7, …, …, 139, 143
BLOCK 4	data codewords d: 4, 8, …, …, …, 364, 368	error correction codewords ε: 4, 8, …, …, …, 140, 144

Figure A.1 — Illustration of interleaving for 72 x 72 symbol

A.2 Starting sequence for interleaving in different sized symbols

The sequence of the interleaved data codewords and error correction codewords is given in Table A.1.

Table A.1 — Sequence of data and error correction codewords for different symbol sizes

Symbol size	Reed-Solomon block	Sequence of data codewords			Sequence of error correction codewords		
52 x 52	1	1, 3, 5	...	201, 203	1, 3, 5	...	81, 83
	2	2, 4, 6	...	202, 204	2, 4, 6	...	82, 84
64 x 64	1	1, 3, 5	...	277, 279	1, 3, 5	...	109, 111
	2	2, 4, 6	...	278, 280	2, 4, 6	...	110, 112
72 x 72	1	1, 5, 9	...	361, 365	1, 5, 9	...	137, 141
	2	2, 6, 10	...	362, 366	2, 6, 10	...	138, 142
	3	3, 7, 11	...	363, 367	3, 7, 11	...	139, 143
	4	4, 8, 12	...	364, 368	4, 8, 12	...	140, 144
80 x 80	1	1, 5, 9	...	449, 453	1, 5, 9	...	185, 189
	2	2, 6, 10	...	450, 454	2, 6, 10	...	186, 190
	3	3, 7, 11	...	451, 455	3, 7, 11	...	187, 191
	4	4, 8, 12	...	452, 456	4, 8, 12	...	188, 192
88 x 88	1	1, 5, 9	...	569, 573	1, 5, 9	...	217, 221
	2	2, 6, 10	...	570, 574	2, 6, 10	...	218, 222
	3	3, 7, 11	...	571, 575	3, 7, 11	...	219, 223
	4	4, 8, 12	...	572, 576	4, 8, 12	...	220, 224
96 x 96	1	1, 5, 9	...	689, 693	1, 5, 9	...	265, 269
	2	2, 6, 10	...	690, 694	2, 6, 10	...	266, 270
	3	3, 7, 11	...	691, 695	3, 7, 11	...	267, 271
	4	4, 8, 12	...	692, 696	4, 8, 12	...	268, 272
104 x 104	1	1, 7, 13	...	805, 811	1, 7, 13	...	325, 331
	2	2, 8, 14	...	806, 812	2, 8, 14	...	326, 332
	3	3, 9, 15	...	807, 813	3, 9, 15	...	327, 333
	4	4, 10, 16	...	808, 814	4, 10, 16	...	328, 334
	5	5, 11, 17	...	809, 815	5, 11, 17	...	329, 335
	6	6, 12, 18	...	810, 816	6, 12, 18	...	330, 336
120 x 120	1	1, 7, 13	...	1039, 1045	1, 7, 13	...	397, 403
	2	2, 8, 14	...	1040, 1046	2, 8, 14	...	398, 404
	3	3, 9, 15	...	1041, 1047	3, 9, 15	...	399, 405
	4	4, 10, 16	...	1042, 1048	4, 10, 16	...	400, 406
	5	5, 11, 17	...	1043, 1049	5, 11, 17	...	401, 407
	6	6, 12, 18	...	1044, 1050	6, 12, 18	...	402, 408

Symbol size	Reed-Solomon block	Sequence of data codewords			Sequence of error correction codewords		
132 x 132	1	1, 9, 17	...	1289, 1297	1, 9, 17	...	481, 489
	2	2, 10, 18	...	1290, 1298	2, 10, 18	...	482, 490
	3	3, 11, 19	...	1291, 1299	3, 11, 19	...	483, 491
	4	4, 12, 20	...	1292, 1300	4, 12, 20	...	484, 492
	5	5, 13, 21	...	1293, 1301	5, 13, 21	...	485, 493
	6	6, 14, 22	...	1294, 1302	6, 14, 22	...	486, 494
	7	7, 15, 23	...	1295, 1303	7, 15, 23	...	487, 495
	8	8, 16, 24	...	1296, 1304	8, 16, 24	...	488, 496
144 x 144	1	1, 11, 21	...	1541, 1551	1, 11, 21	...	601, 611
	2	2, 12, 22	...	1542, 1552	2, 12, 22	...	602, 612
	3	3, 13, 23	...	1543, 1553	3, 13, 23	...	603, 613
	4	4, 14, 24	...	1544, 1554	4, 14, 24	...	604, 614
	5	5, 15, 25	...	1545, 1555	5, 15, 25	...	605, 615
	6	6, 16, 26	...	1546, 1556	6, 16, 26	...	606, 616
	7	7, 17, 27	...	1547, 1557	7, 17, 27	...	607, 617
	8	8, 18, 28	...	1548, 1558	8, 18, 28	...	608, 618
	9	9, 19, 29	...	1549	9, 19, 29	...	609, 619
	10	10, 20, 30	...	1550	10, 20, 30	...	610, 620

Annex B
(normative)

ECC 200 pattern randomising

The pattern randomising algorithms convert an input codeword at a given position to a new randomised output codeword.

B.1 253-state algorithm

This algorithm adds a pseudo-random number to the Pad codeword value. The pseudo-random number will always be in the range 1 to 253 and the randomised Pad codeword value will be in the range 1 to 254.

The variable Pad_codeword_position is the number of data codewords from the beginning of encoded data.

B.1.1 253-state randomising algorithm

```
INPUT ( Pad_codeword_value, Pad_codeword_position )

pseudo_random_number = ( ( 149 * Pad_codeword_position ) mod 253 ) + 1

temp_variable = Pad_codeword_value + pseudo_random_number

IF ( temp_variable <= 254 )

    OUTPUT ( randomised_Pad_codeword_value = temp_variable )

ELSE

    OUTPUT ( randomised_Pad_codeword_value = temp_variable - 254 )
```

B.1.2 253-state un-randomising algorithm

```
INPUT ( randomised_Pad_codeword_value, Pad_codeword_position )

pseudo_random_number = ( ( 149 * Pad_codeword_position ) mod 253 ) + 1

temp_variable = randomised_Pad_codeword_value - pseudo_random_number

IF ( temp_variable >= 1 )

    OUTPUT ( Pad_codeword_value = temp_variable)

ELSE

    OUTPUT ( Pad_codeword_value = temp_variable + 254 )
```

B.2 255-state algorithm

This algorithm adds a pseudo-random number to the Base 256 encodation codeword value. The pseudorandom number will always be in the range 1 to 255 and the randomised Base 256 codeword value will be in the range 0 to 255.

The variable Base256_codeword_position is the number of data codewords from the beginning of encoded data.

B.2.1 255-state randomising algorithm

```
INPUT ( Base256_codeword_value, Base256_codeword_position )

pseudo_random_number = ( ( 149 * Base256_codeword_position ) mod 255 ) + 1

temp_variable = Base256_codeword_value + pseudo_random_number

IF ( temp_variable <= 255 )

    OUTPUT (randomised_Base256_codeword_value = temp_variable )

ELSE

    OUTPUT (randomised_Base256_codeword_value = temp_variable - 256 )
```

B.2.2 255-state un-randomising algorithm

```
INPUT ( randomised_Base256_codeword_value, Base256_codeword_position )

pseudo_random_number = ( ( 149 * Base256_codeword_position ) mod 255 ) + 1

temp_variable=randomised_Base256_codeword_value - pseudo_random_number

IF ( temp_variable >= 0 )

    OUTPUT ( Base256_codeword_value = temp_variable )

ELSE

    OUTPUT ( Base256_codeword_value = temp_variable + 256 )
```

Annex C
(normative)

ECC 200 encodation character sets

Table C.1 — C40 encodation character set

C40 Value	Basic set		Shift 1 set		Shift 2 set		Shift 3 set	
	Char	Decimal	Char	Decimal	Char	Decimal	Char	Decimal
0	Shift 1		NUL	0	!	33	‘	96
1	Shift 2		SOH	1	“	34	a	97
2	Shift 3		STX	2	#	35	b	98
3	space	32	ETX	3	$	36	c	99
4	0	48	EOT	4	%	37	d	100
5	1	49	ENQ	5	&	38	e	101
6	2	50	ACK	6	‘	39	f	102
7	3	51	BEL	7	(	40	g	103
8	4	52	BS	8	)	41	h	104
9	5	53	HT	9	*	42	i	105
10	6	54	LF	10	+	43	j	106
11	7	55	VT	11	,	44	k	107
12	8	56	FF	12	-	45	l	108
13	9	57	CR	13	.	46	m	109
14	A	65	SO	14	/	47	n	110
15	B	66	SI	15	:	58	o	111
16	C	67	DLE	16	;	59	p	112
17	D	68	DC1	17	<	60	q	113
18	E	69	DC2	18	=	61	r	114
19	F	70	DC3	19	>	62	s	115
20	G	71	DC4	20	?	63	t	116
21	H	72	NAK	21	@	64	u	117
22	I	73	SYN	22	[	91	v	118
23	J	74	ETB	23	\	92	w	119
24	K	75	CAN	24	]	93	x	120
25	L	76	EM	25	^	94	y	121
26	M	77	SUB	26	_	95	z	122
27	N	78	ESC	27	FNC1		{	123
28	O	79	FS	28			|	124
29	P	80	GS	29			}	125
30	Q	81	RS	30	Upper Shift		~	126
31	R	82	US	31			DEL	127
32	S	83						
33	T	84						
34	U	85						
35	V	86						

C40 Value	Basic set		Shift 1 set		Shift 2 set		Shift 3 set	
	Char	Decimal	Char	Decimal	Char	Decimal	Char	Decimal
36	W	87						
37	X	88						
38	Y	89						
39	Z	90						

NOTE The relationship between the ASCII decimal value and the C40 value remains constant regardless of which ECI is in effect.

Table C.2 — Text encodation character set

Text value	Basic set		Shift 1 set		Shift 2 set		Shift 3 set	
	Char	Decimal	Char	Decimal	Char	Decimal	Char	Decimal
0	Shift	1	NUL	0	!	33	‘	96
1	Shift	2	SOH	1	“	34	A	65
2	Shift	3	STX	2	#	35	B	66
3	space	32	ETX	3	$	36	C	67
4	0	48	EOT	4	%	37	D	68
5	1	49	ENQ	5	&	38	E	69
6	2	50	ACK	6	‘	39	F	70
7	3	51	BEL	7	(	40	G	71
8	4	52	BS	8	)	41	H	72
9	5	53	HT	9	*	42	I	73
10	6	54	LF	10	+	43	J	74
11	7	55	VT	11	,	44	K	75
12	8	56	FF	12	-	45	L	76
13	9	57	CR	13	.	46	M	77
14	a	97	SO	14	/	47	N	78
15	b	98	SI	15	:	58	O	79
16	c	99	DLE	16	;	59	P	80
17	d	100	DC1	17	<	60	Q	81
18	e	101	DC2	18	=	61	R	82
19	f	102	DC3	19	>	62	S	83
20	g	103	DC4	20	?	63	T	84
21	h	104	NAK	21	@	64	U	85
22	i	105	SYN	22	[	91	V	86
23	j	106	ETB	23	\	92	W	87
24	k	107	CAN	24	]	93	X	88
25	l	108	EM	25	^	94	Y	89
26	m	109	SUB	26	_	95	Z	90
27	n	110	ESC	27	FNC1		{	123
28	o	111	FS	28			|	124
29	p	112	GS	29			}	125
30	q	113	RS	30	Upper	Shift	~	126
31	r	114	US	31			DEL	127
32	s	115						
33	t	116						
34	u	117						
35	v	118						
36	w	119						
37	x	120						

Text value	Basic set		Shift 1 set		Shift 2 set		Shift 3 set	
	Char	Decimal	Char	Decimal	Char	Decimal	Char	Decimal
38	y	121						
39	z	122						

NOTE The relationship between the ASCII decimal value and the Text value remains constant regardless of which ECI is in effect.

Table C.3 — EDIFACT encodation character set

Data character			EDIFACT binary value	Data character			EDIFACT binary value
Char	Decimal value	Binary value		Char	Decimal value	Binary value	
@	64	01000000	000000	space	32	00100000	100000
A	65	01000001	000001	!	33	00100001	100001
B	66	01000010	000010	"	34	00100010	100010
C	67	01000011	000011	#	35	00100011	100011
D	68	01000100	000100	$	36	00100100	100100
E	69	01000101	000101	%	37	00100101	100101
F	70	01000110	000110	&	38	00100110	100110
G	71	01000111	000111	'	39	00100111	100111
H	72	01001000	001000	(	40	00101000	101000
I	73	01001001	001001	)	41	00101001	101001
J	74	01001010	001010	*	42	00101010	101010
K	75	01001011	001011	+	43	00101011	101011
L	76	01001100	001100	,	44	00101100	101100
M	77	01001101	001101	-	45	00101101	101101
N	78	01001110	001110	.	46	00101110	101110
O	79	01001111	001111	/	47	00101111	101111
P	80	01010000	010000	0	48	00110000	110000
Q	81	01010001	010001	1	49	00110001	110001
R	82	01010010	010010	2	50	00110010	110010
S	83	01010011	010011	3	51	00110011	110011
T	84	01010100	010100	4	52	00110100	110100
U	85	01010101	010101	5	53	00110101	110101
V	86	01010110	010110	6	54	00110110	110110
W	87	01010111	010111	7	55	00110111	110111
X	88	01011000	011000	8	56	00111000	111000
Y	89	01011001	011001	9	57	00111001	111001
Z	90	01011010	011010	:	58	00111010	111010
[	91	01011011	011011	;	59	00111011	111011
\	92	01011100	011100	<	60	00111100	111100
]	93	01011101	011101	=	61	00111101	111101
^	94	01011110	011110	>	62	00111110	111110
Unlatch		01011111	011111	?	63	00111111	111111

NOTE The relationship between the ASCII decimal value and the EDIFACT value remain constant regardless of which ECI is in effect.

Annex D
(normative)

ECC 200 alignment patterns

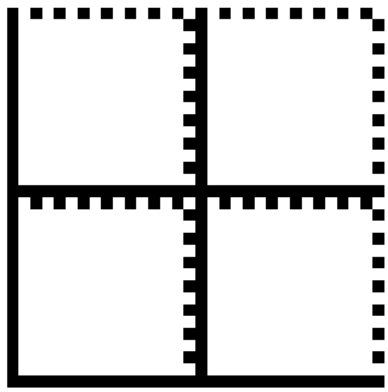


Figure D.1 — Alignment pattern configuration for 32 x 32 square symbol

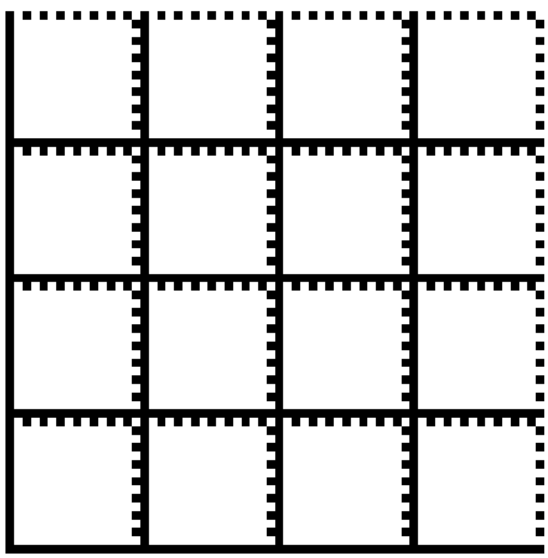


Figure D.2 — Alignment pattern configuration for 64 x 64 square symbol

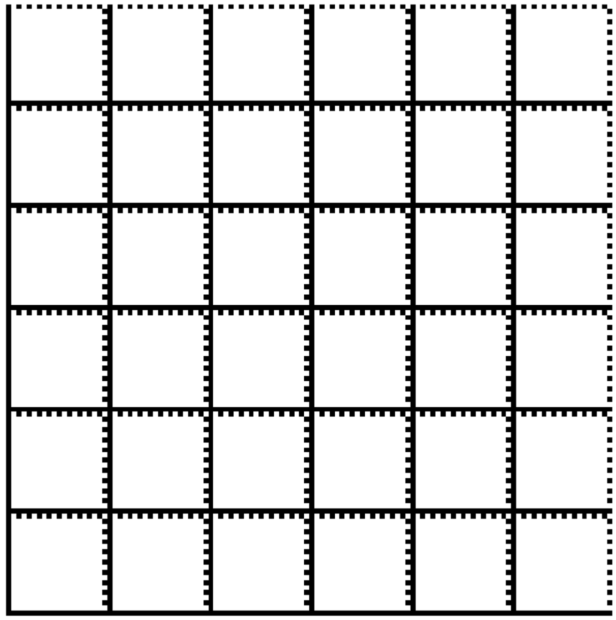


Figure D.3 — Alignment pattern configuration for 120 x 120 square symbol

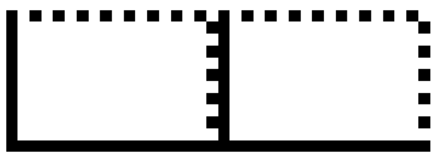


Figure D.4 — Alignment pattern configuration for 12 x 36 rectangular symbol

Annex E
(normative)

ECC 200 Reed-Solomon error detection and correction

E.1 Error correction codeword generator polynomials

The error correction codewords are the coefficients of the remainder resulting from first multiplying the symbol data polynomial d(x) by x^k and then dividing it by the generator polynomial g(x). Each generator polynomial is the product of the first-degree polynomials: $x - 2^1$, $x - 2^2$, ..., $x - 2^n$; where n is the degree of the generator polynomial.

For example the fifth degree generator polynomial is:

$(x + 2)(x + 4)(x + 8)(x + 16)(x + 32)$

$= x^5 + (2 + 4 + 8 + 16 + 32)x^4 + ((2 * 4) + (2 * 8) + (2 * 16) + (2 * 32) + (4 * 8) + (4 * 16) + (4 * 32) + (8 * 16) + (8 * 32) + (16 * 32))x^3 + ((2 * 4 * 8) + (2 * 4 * 16) + (2 * 4 * 32) + (2 * 8 * 16) + (2 * 8 * 32) + (2 * 16 * 32) + (4 * 8 * 16) + (4 * 8 * 32) + (4 * 16 * 32) + (8 * 16 * 32))x^2 + ((2 * 4 * 8 * 16) + (2 * 4 * 8 * 32) + (2 * 4 * 16 * 32) + (2 * 8 * 16 * 32) + (4 * 8 * 16 * 32))x + (2 * 4 * 8 * 16 * 32)$

$= x^5 + 62x^4 + 111x^3 + 15x^2 + 48x + 228.$

Note that this Galois Field arithmetic is not normal integer arithmetic: - is equivalent to +, which is an “exclusive-or” operation in this Field, and multiplication is byte-wise modulo 100101101 for each binary polynomial term generated by bit-by-bit multiplication.

The polynomial divisor for generating 5 check characters is:

$g(x) = x^5 + 62x^4 + 111x^3 + 15x^2 + 48x + 228.$

The polynomial divisor for generating 7 check characters is:

$g(x) = x^7 + 254x^6 + 92x^5 + 240x^4 + 134x^3 + 144x^2 + 68x + 23.$

The polynomial divisor for generating 10 check characters is:

$g(x) = x^{10} + 61x^9 + 110x^8 + 255x^7 + 116x^6 + 248x^5 + 223x^4 + 166x^3 + 185x^2 + 24x + 28.$

The polynomial divisor for generating 11 check characters is:

$g(x) = x^{11} + 120x^{10} + 97x^9 + 60x^8 + 245x^7 + 39x^6 + 168x^5 + 194x^4 + 12x^3 + 205x^2 + 138x + 175.$

The polynomial divisor for generating 12 check characters is:

$g(x) = x^{12} + 242x^{11} + 100x^{10} + 178x^9 + 97x^8 + 213x^7 + 142x^6 + 42x^5 + 61x^4 + 91x^3 + 158x^2 + 153x + 41.$

The polynomial divisor for generating 14 check characters is:

$g(x) = x^{14} + 185x^{13} + 83x^{12} + 186x^{11} + 18x^{10} + 45x^9 + 138x^8 + 119x^7 + 157x^6 + 9x^5 + 95x^4 + 252x^3 + 192x^2 + 97x + 156.$

The polynomial divisor for generating 18 check characters is:

$$g(x) = x^{18} + 188x^{17} + 90x^{16} + 48x^{15} + 225x^{14} + 254x^{13} + 94x^{12} + 129x^{11} + 109x^{10} + 213x^{9} + 241x^{8} + 61x^{7} + 66x^{6} + 75x^{5} + 188x^{4} + 39x^{3} + 100x^{2} + 195x + 83.$$

The polynomial divisor for generating 20 check characters is:

$$g(x) = x^{20} + 172x^{19} + 186x^{18} + 174x^{17} + 27x^{16} + 82x^{15} + 108x^{14} + 79x^{13} + 253x^{12} + 145x^{11} + 153x^{10} + 160x^{9} + 188x^{8} + 2x^{7} + 168x^{6} + 71x^{5} + 233x^{4} + 9x^{3} + 244x^{2} + 195x + 15.$$

The polynomial divisor for generating 24 check characters is:

$$g(x) = x^{24} + 193x^{23} + 50x^{22} + 96x^{21} + 184x^{20} + 181x^{19} + 12x^{18} + 124x^{17} + 254x^{16} + 172x^{15} + 5x^{14} + 21x^{13} + 155x^{12} + 223x^{11} + 251x^{10} + 197x^{9} + 155x^{8} + 21x^{7} + 176x^{6} + 39x^{5} + 109x^{4} + 205x^{3} + 88x^{2} + 190x + 52.$$

The polynomial divisor for generating 28 check characters is:

$$g(x) = x^{28} + 255x^{27} + 93x^{26} + 168x^{25} + 233x^{24} + 151x^{23} + 120x^{22} + 136x^{21} + 141x^{20} + 213x^{19} + 110x^{18} + 138x^{17} + 17x^{16} + 121x^{15} + 249x^{14} + 34x^{13} + 75x^{12} + 53x^{11} + 170x^{10} + 151x^{9} + 37x^{8} + 174x^{7} + 103x^{6} + 96x^{5} + 71x^{4} + 97x^{3} + 43x^{2} + 231x + 211.$$

The polynomial divisor for generating 36 check characters is:

$$g(x) = x^{36} + 112x^{35} + 81x^{34} + 98x^{33} + 225x^{32} + 25x^{31} + 59x^{30} + 184x^{29} + 175x^{28} + 44x^{27} + 115x^{26} + 119x^{25} + 95x^{24} + 137x^{23} + 101x^{22} + 33x^{21} + 68x^{20} + 4x^{19} + 2x^{18} + 18x^{17} + 229x^{16} + 182x^{15} + 80x^{14} + 251x^{13} + 220x^{12} + 179x^{11} + 84x^{10} + 120x^{9} + 102x^{8} + 181x^{7} + 162x^{6} + 250x^{5} + 130x^{4} + 218x^{3} + 242x^{2} + 127x + 245.$$

The polynomial divisor for generating 42 check characters is:

$$g(x) = x^{42} + 5x^{41} + 9x^{40} + 5x^{39} + 226x^{38} + 177x^{37} + 150x^{36} + 50x^{35} + 69x^{34} + 202x^{33} + 248x^{32} + 101x^{31} + 54x^{30} + 57x^{29} + 253x^{28} + x^{27} + 21x^{26} + 121x^{25} + 57x^{24} + 111x^{23} + 214x^{22} + 105x^{21} + 167x^{20} + 9x^{19} + 100x^{18} + 95x^{17} + 175x^{16} + 8x^{15} + 242x^{14} + 133x^{13} + 245x^{12} + 2x^{11} + 122x^{10} + 105x^{9} + 247x^{8} + 153x^{7} + 22x^{6} + 38x^{5} + 19x^{4} + 31x^{3} + 137x^{2} + 193x + 77.$$

The polynomial divisor for generating 48 check characters is:

$$g(x) = x^{48} + 19x^{47} + 225x^{46} + 253x^{45} + 92x^{44} + 213x^{43} + 69x^{42} + 175x^{41} + 160x^{40} + 147x^{39} + 187x^{38} + 87x^{37} + 176x^{36} + 44x^{35} + 82x^{34} + 240x^{33} + 186x^{32} + 138x^{31} + 66x^{30} + 100x^{29} + 120x^{28} + 88x^{27} + 131x^{26} + 205x^{25} + 170x^{24} + 90x^{23} + 37x^{22} + 23x^{21} + 118x^{20} + 147x^{19} + 16x^{18} + 106x^{17} + 191x^{16} + 87x^{15} + 237x^{14} + 188x^{13} + 205x^{12} + 231x^{11} + 238x^{10} + 133x^{9} + 238x^{8} + 22x^{7} + 117x^{6} + 32x^{5} + 96x^{4} + 223x^{3} + 172x^{2} + 132x + 245.$$

The polynomial divisor for generating 56 check characters is:

$$g(x) = x^{56} + 46x^{55} + 143x^{54} + 53x^{53} + 233x^{52} + 107x^{51} + 203x^{50} + 43x^{49} + 155x^{48} + 28x^{47} + 247x^{46} + 67x^{45} + 127x^{44} + 245x^{43} + 137x^{42} + 13x^{41} + 164x^{40} + 207x^{39} + 62x^{38} + 117x^{37} + 201x^{36} + 150x^{35} + 22x^{34} + 238x^{33} + 144x^{32} + 232x^{31} + 29x^{30} + 203x^{29} + 117x^{28} + 234x^{27} + 218x^{26} + 146x^{25} + 228x^{24} + 54x^{23} + 132x^{22} + 200x^{21} + 38x^{20} + 223x^{19} + 36x^{18} + 159x^{17} + 150x^{16} + 235x^{15} + 215x^{14} + 192x^{13} + 230x^{12} + 170x^{11} + 175x^{10} + 29x^{9} + 100x^{8} + 208x^{7} + 220x^{6} + 17x^{5} + 12x^{4} + 238x^{3} + 223x^{2} + 9x + 175.$$

The polynomial divisor for generating 62 check characters is:

$$g(x) = x^{62} + 204x^{61} + 11x^{60} + 47x^{59} + 86x^{58} + 124x^{57} + 224x^{56} + 166x^{55} + 94x^{54} + 7x^{53} + 232x^{52} + 107x^{51} + 4x^{50} + 170x^{49} + 176x^{48} + 31x^{47} + 163x^{46} + 17x^{45} + 188x^{44} + 130x^{43} + 40x^{42} + 10x^{41} + 87x^{40} + 63x^{39} + 51x^{38} + 218x^{37} + 27x^{36} + 6x^{35} + 147x^{34} + 44x^{33} + 161x^{32} + 71x^{31} + 114x^{30} + 64x^{29} + 175x^{28} + 221x^{27} + 185x^{26} + 106x^{25} + 250x^{24} + 190x^{23} + 197x^{22} + 63x^{21} + 245x^{20} + 230x^{19} + 134x^{18} + 112x^{17} + 185x^{16} + 37x^{15} + 196x^{14} + 108x^{13} + 143x^{12} + 189x^{11} + 201x^{10} + 188x^{9} + 202x^{8} + 118x^{7} + 39x^{6} + 210x^{5} + 144x^{4} + 50x^{3} + 169x^{2} + 93x + 242.$$

The polynomial divisor for generating 68 check characters is:

$g(x) = x^{68} + 186x^{67} + 82x^{66} + 103x^{65} + 96x^{64} + 63x^{63} + 132x^{62} + 153x^{61} + 108x^{60} + 54x^{59} + 64x^{58} + 189x^{57} + 211x^{56} + 232x^{55} + 49x^{54} + 25x^{53} + 172x^{52} + 52x^{51} + 59x^{50} + 241x^{49} + 181x^{48} + 239x^{47} + 223x^{46} + 136x^{45} + 231x^{44} + 210x^{43} + 96x^{42} + 232x^{41} + 220x^{40} + 25x^{39} + 179x^{38} + 167x^{37} + 202x^{36} + 185x^{35} + 153x^{34} + 139x^{33} + 66x^{32} + 236x^{31} + 227x^{30} + 160x^{29} + 15x^{28} + 213x^{27} + 93x^{26} + 122x^{25} + 68x^{24} + 177x^{23} + 158x^{22} + 197x^{21} + 234x^{20} + 180x^{19} + 248x^{18} + 136x^{17} + 213x^{16} + 127x^{15} + 73x^{14} + 36x^{13} + 154x^{12} + 244x^{11} + 147x^{10} + 33x^{9} + 89x^{8} + 56x^{7} + 159x^{6} + 149x^{5} + 251x^{4} + 89x^{3} + 173x^{2} + 228x + 220$.

E.2 Error correction calculation

The Peterson-Gorenstein-Zierler algorithm may be used to correct errors in decoded ECC 200 symbols.

The calculation described below follows this error correcting algorithm, using the Reed-Solomon error correction codewords.

Erasures shall be corrected as errors by initially filling any erasure codeword positions with dummy values.

All calculations shall be done using GF(2^8) arithmetic operations. Addition and subtraction are equivalent to the binary XOR operation. Multiplication and division can be performed using log and antilog tables.

Construct the symbol character polynomial $C(x) = C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + \ldots + C_1x^1 + C_0$ where the n coefficients are the codewords read with C_{n-1} being the first symbol character and where n is the total number of symbol characters.

Calculate i syndrome values S_0 through S_{i-1} by evaluating $C(x)$ at $x = 2^k$ for $k = 1$ through i, where i is the number of error correction codewords in the symbol.

Form and solve j simultaneous equations with j unknowns L_0 through L_{j-1} using the i syndromes:

$$S_0L_0 + S_1L_1 + \ldots + S_{j-1}L_{j-1} = S_j$$

$$S_1L_0 + S_2L_1 + \ldots + S_jL_{j-1} = S_{j+1}$$

$$\vdots$$

$$\vdots$$

$$S_{j-1}L_0 + S_jL_1 + \ldots + S_{2j-2}L_{j-1} = S_{2j-1}$$

where j is $i/2$.

Construct the error locator polynomial:

$$L(x) = L_{j-1}x^j + L_{j-2}x^{j-1} + \ldots + L_0x + 1$$

from the j values of L obtained above. Evaluate $L(x)$ at $x = 2^k$ for $k = 0$ through $n - 1$ where n is the total number of symbol characters in the symbol.

Whenever $L(2^k) = 0$, an error location is given by $n - 1 - k$. If more than j error locations are found, the symbol is not correctable.

Save the error locations in m error location variables E_0 through E_{m-1} where m is the number of error locations found. Form and solve m simultaneous equations with m unknowns X_0 through X_{m-1} (the error magnitudes) using the error location variables E and the first m syndromes S:

$$E_0X_0 + E_1X_1 + \ldots + E_{m-1}X_{m-1} = S_0$$

$$E_0^2X_0 + E_1^2X_1 + \ldots + E_{(m-1)}^2X_{m-1} = S_1$$

$$E_0^3X_0 + E_1^3X_1 + \ldots + E_{(m-1)}^3X_{m-1} = S_2$$

:

:

$$E_0^mX_0 + E_1^mX_1 + \ldots + E_{(m-1)}^mX_{m-1} = S_{m-1}$$

Add the error magnitudes X_0 through $X_{m-}1$ to the symbol character values at the corresponding error locations E_0 through E_{m-1} to correct the errors.

NOTE E_0 …. E_{m-1} – are the roots of the error locator polynomial.

This algorithm, written in C, is available from AIM, Inc. on the Data Matrix Developers Diskette (see Bibliography).

E.3 Calculation of error correction codewords

The following is an example of a generic routine, written in C, which calculates the error correction codewords for a given data codeword string of length "nd", stored as an integer array wd[]. The function ReedSolomon() first generates log and antilog tables for the Galois Field of size "gf" (in the case of ECC 200, 2^8) with prime modulus "pp" (in the case of ECC 200, 301), then uses them in the function prod(), first to calculate coefficients of the generator polynomial of order "nc" and then to calculate "nc" additional check codewords which are appended to the data in wd[].

```
/* "prod(x,y,log,alog,gf)" returns the product "x" times "y" */
int prod(int x, int y, int *log, int *alog, int gf) {
   if (!x || !y) return 0;
   ELSE return alog[(log[x] + log[y]) % (gf-1)];
}

/* "ReedSolomon(wd,nd,nc,gf.pp)" takes "nd" data codeword values in wd[] */
/* and adds on "nc" check codewords, all within GF(gf) where "gf" is a */
/* power of 2 and "pp" is the value of its prime modulus polynomial */
void ReedSolomon(int *wd, int nd, int nc, int gf, int pp) {
   int i, j, k. *log,*alog,*c;

/* allocate, then generate the log & antilog arrays: */
   log = malloc(sizeof(int) * gf);
   alog = malloc(sizeof(int) * gf);
   log[0] = 1-gf; alog[0] = 1;
   for (i = 1; i < gf; i++) {
      alog[i] = alog[i-1] * 2;
      if (alog[i] >= gf) alog[i] ^= pp;
      log[alog[i]] = i;
}

/* allocate, then generate the generator polynomial coefficients: */
   c = malloc(sizeof(int) * (nc+1));
   for (i=1; i<=nc; i++) c[i] = 0; c[0] = 1;
```

```
    for (i=1; i<=nc; i++) {
        c[i] = c[i-1];
        for (j=i-1; j>=1; j--) {
            c[j] = c[j-1] ^ prod(c[j],alog[i],log,alog,gf);
        }
        c[0] = prod(c[0],alog[i],log,alog,gf);
    }

/* clear, then generate "nc" checkwords in the array wd[] : */
    for (i=nd; i<=(nd+nc); i++) wd[i] = 0;
    for (i=0; i<nd; i++) {
        k = wd[nd] ^ wd[i] ;
        for (j=0; j<nc; j++) {
            wd[nd+j] = wd[nd+j+1] ^ prod(k,c[nc-j-1],log, alog,gf);
        }
    }

    free(c);
    free(alog);
    free(log);
}
```

Annex F
(normative)

ECC 200 symbol character placement

F.1 Symbol character placement

The following C language program generates symbol character placement diagrams:

```
#include <stdio.h>
#include <alloc.h>

int nrow, ncol, *array;

/* "module" places "chr+bit" with appropriate wrapping within array[] */
void module(int row, int col, int chr, int bit)
{ if (row < 0) { row += nrow; col += 4 - ((nrow+4)%8); }
   if (col < 0) { col += ncol; row += 4 - ((ncol+4)%8); }
   array[row*ncol+col] = 10*chr + bit;
}
/* "utah" places the 8 bits of a utah-shaped symbol character in ECC200 */
void utah(int row, int col, int chr)
{ module(row-2,col-2,chr,1);
   module(row-2,col-1,chr,2);
   module(row-1,col-2,chr,3);
   module(row-1,col-1,chr,4);
   module(row-1,col,chr,5);
   module(row,col-2,chr,6);
   module(row,col-1,chr,7);
   module(row,col,chr,8);
}
/* "cornerN" places 8 bits of the four special corner cases in ECC200 */
void corner1(int chr)
{ module(nrow-1,0,chr,1);
   module(nrow-1,1,chr,2);
   module(nrow-1,2,chr,3);
   module(0,ncol-2,chr,4);
   module(0,ncol-1,chr,5);
   module(1,ncol-1,chr,6);
   module(2,ncol-1,chr,7);
   module(3,ncol-1,chr,8);
}
void corner2(int chr)
{ module(nrow-3,0,chr,1);
   module(nrow-2,0,chr,2);
   module(nrow-1,0,chr,3);
   module(0,ncol-4,chr,4);
   module(0,ncol-3,chr,5);
   module(0,ncol-2,chr,6);
   module(0,ncol-1,chr,7);
   module(1,ncol-1,chr,8);
}
void corner3(int chr)
{ module(nrow-3,0,chr,1);
   module(nrow-2,0,chr,2);
   module(nrow-1,0,chr,3);
   module(0,ncol-2,chr,4);
   module(0,ncol-1,chr,5);
   module(1,ncol-1,chr,6);
```

```
   module(2,ncol-1,chr,7);
   module(3,ncol-1,chr,8);
}
void corner4(int chr)
{ module(nrow-1,0,chr,1);
   module(nrow-1,ncol-1,chr,2);
   module(0,ncol-3,chr,3);
   module(0,ncol-2,chr,4);
   module(0,ncol-1,chr,5);
   module(1,ncol-3,chr,6);
   module(1,ncol-2,chr,7);
   module(1,ncol-1,chr,8);
}
/* "ECC200" fills an nrow x ncol array with appropriate values for ECC200 */
void ECC200(void)
{ int row, col, chr;

/* First, fill the array[] with invalid entries */
   for (row=0; row<nrow; row++) {
      for (col=0; col<ncol; col++) {
         array[row*ncol+col] = 0;
      }
   }
/* Starting in the correct location for character #1, bit 8,... */
   chr = 1; row = 4; col = 0;

   do {
/* repeatedly first check for one of the special corner cases, then... */
      if ((row == nrow) && (col == 0)) corner1(chr++);
      if ((row == nrow-2) && (col == 0) && (ncol%4)) corner2(chr++);
      if ((row == nrow-2) && (col == 0) && (ncol%8 == 4)) corner3(chr++);
      if ((row == nrow+4) && (col == 2) && (!(ncol%8))) corner4(chr++);
/* sweep upward diagonally, inserting successive characters,... */
   do {
      if ((row < nrow) && (col >= 0) && (!array[row*ncol+col]))
         utah(row,col,chr++);
      row -= 2; col += 2;
   } while ((row >= 0) && (col < ncol));
   row += 1; col += 3;

/* & then sweep downward diagonally, inserting successive characters,... */
   +
   do {
      if ((row >= 0) && (col < ncol) && (!array[row*ncol+col]))
         utah(row,col,chr++);
      row += 2; col -= 2;
   } while ((row < nrow) && (col >= 0));
   row += 3; col += 1;

/* ... until the entire array is scanned */
   } while ((row < nrow) || (col < ncol));

/* Lastly, if the lower righthand corner is untouched, fill in fixed pattern */
   if (!array[nrow*ncol-1]) {
      array[nrow*ncol-1] = array[nrow*ncol-ncol-2] = 1;
   }
}

/* "main" checks for valid command line entries, then computes & displays array
*/
void main(int argc, char *argv[])
{ int x, y, z;

   if (argc =< 3) {
      printf("Command line: ECC200 #_of_Data_Rows #_of_Data_Columns\n");
```

```
    } ELSE {
        nrow = ncol = 0;
        nrow = atoi(argv[1]); ncol = atoi(argv[2]);
        if ((nrow >= 6) && (~nrow&0x01) && (ncol >= 6) && (~ncol&0x01)) {
            array = malloc(sizeof(int) * nrow * ncol);

            ECC200();

            for (x=0; x<nrow; x++) {
                for (y=0; y<ncol; y++) {
                    z = array[x*ncol+y];
                    if (z == 0) printf(" WHI");
                        ELSE if (z == 1) printf("BLK");
                            ELSE printf("%3d.%d",z/10,z%10);
                }
                printf("\n");
            }
            free(array);
        }
    }
}
```

F.2 Symbol character placement rules

F.2.1 Non-standard symbol character shapes

Because the standard symbol character shape cannot always fit at the data module boundaries of the symbol and at some corners, a small set of non-standard symbol characters is required. There are six conditions: two boundary conditions which affect all symbol formats, and four different corner conditions which apply to certain symbol formats:

a. One portion of the symbol character shape is placed on one side and the other on the opposite side. This applies to two basic symbol character shapes (see Figure F.1). Variants of these arrangements concern the row-to-row relationship between the left and right hand boundary (see Table F.1).

b. One portion of the symbol character is placed on the top boundary and the other portion on the bottom boundary. This applies to two basic symbol character shapes (see Figure F.2). Variants of these arrangements concern the column-to-column relationship between the top and bottom boundary (see Table F.1).

c. Four symbol character shapes are split between two or three corners (see Figures F.3 to F.6) The non-standard symbol shapes are placed at opposite boundaries. The number of these pairings increases in general proportion to the size of the perimeter of the mapping matrix. The basic pattern is as illustrated in Figures F.1 and F.2. In Figure F.1, modules a8 and a7 are in the same row, as are modules b7 and b6. In Figure F.2 module c6 and c3 are in the same column as are modules d3 and d1. There are seven cases for boundary placement, which define the relative vertical position of the symbol characters illustrated in Figure F.1, the horizontal position of the symbol characters illustrated in Figure F.2, and the corner conditions.

Table F.1 — Factors which determine the boundary placement cases

Boundary placement case	Row relationship of module a8 and a7	Column relationship of module c6 and c3	Corner condition Figure No.	Mapping matrices affected	Refer to Annex F. Figure no. for example
1	a7 Row = a8 Row	c3 Column = c6 Column	None	Square: 8^2, 16^2, 24^2, 32^2, 40^2 48^2, 56^2, 64^2, 72^2, 80^2, 88^2, 96^2, & 120^2	Figure F.9 & F.16
2	a7 Row = a8 Row - 2	c3 Column = c6 Column - 2	None	Square: 10^2 & 18^2	Figure F.10 & F.17
3	a7 Row = a8 Row + 4	c3 Column = c6 Column + 4	F.3	Square: 12^2, 20^2,28^2, 36^2, 44^2, 108^2, & 132^2	Figure F.11 & F.18
4	a7 Row = a8 Row + 2	c3 Column = c6 Column + 2	F.4	Square: 14^2 &22^2	Figure F.12 & F.19
5	a7 Row = a8 Row	c3 Column = c6 Column + 2	F.5	Rectangular: 6 x 16 & 14 x 32	Figure F.13
6	a7 Row = a8 Row	c3 Column = c6 Column - 2	None	Rectangular: 10 x 24 & 10 x 32	Figure F.14
7	a7 Row = a8 Row + 4	c3 Column = c6 Column + 2	F.6	Rectangular: 6 x 28 & 14 x 44	Figure F.15

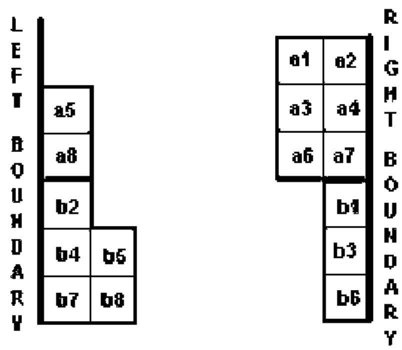


Figure F.1 — Left and right symbol characters

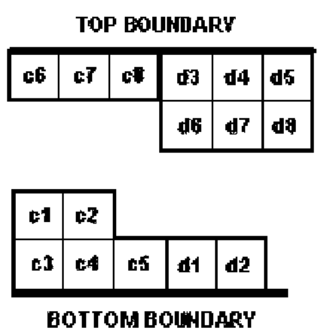


Figure F.2 — Top and bottom symbol characters

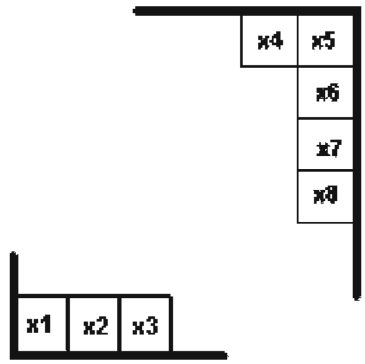


Figure F.3 — Corner condition 1

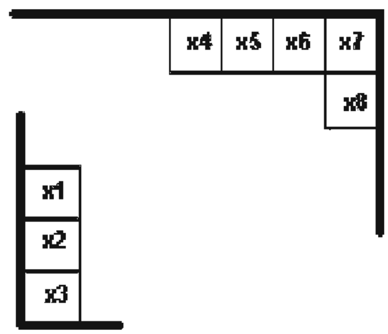


Figure F.4 — Corner condition 2

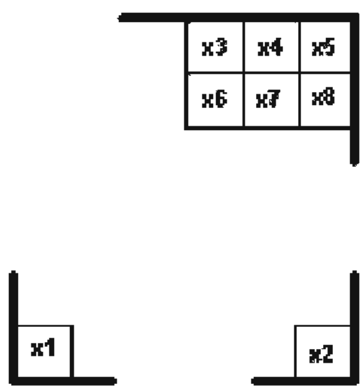


Figure F.5 — Corner condition 3

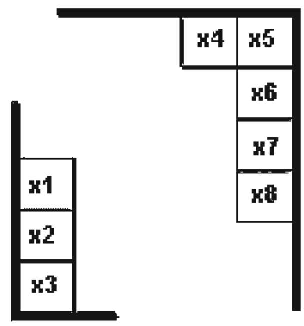


Figure F.6 — Corner condition 4

NOTE 1 Algebraic notation has been used to identify the symbol characters because these vary depending on the symbol format

NOTE 2 The corner characters are identified by the module in the bottom left and top right corners.

F.2.2 Symbol character arrangement

The symbol characters are placed in a matrix in the following manner:

a) A mapping matrix is created.

   1) For small symbols with only one data region, this equates to the mapping matrix.

   2) For larger symbols with more than one data region, the mapping matrix equates to an area the size of the abutted data regions. In effect, the mapping matrix has no separating alignment patterns. For example, the 36 x 36 format symbol has four 16 x 16 data regions which abut to create a mapping matrix 32 x 32. The size of the mapping matrix for each symbol format is given in Table 7. The boundary placement case is given in Table F.1.

b) Symbol character 2 is placed in the uppermost left position, with its modules conforming to the bit (or module) sequence defined in Figure 11. Using the notation 2.1 to identify module 1 of symbol character 2, this module is in the top row and leftmost column of every mapping matrix. The module array sequence shown in Figure F.7 is constant for all mapping matrices.

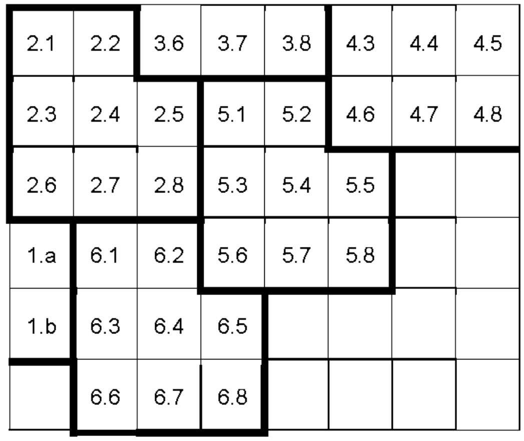


Figure F.7 — Starting sequence for module placement

NOTE The values a and b depend on the size of the mapping matrix

c) The corner shapes are positioned according to Table F.1 and the appropriate Figures F.3 to F.6. Plotting of the standard symbol character shapes continues, nesting the shapes as illustrated above for symbol characters 2, 5, and 6. The non-standard symbol characters are positioned as per Table F.1. This process results in the mapping matrix being completely covered in symbol characters, most of which are un-numbered.

d) The sequence of symbol characters is determined as follows. Symbol characters are arranged on 45-degree parallel diagonal lines between the lower left and upper right, generally linking through the centres on module 8.

e) The first diagonal line starts with the line through module 8 of symbol character 1; this is module 8 except in the case of the 6 x 28 mapping matrix, where the corner condition, as defined in Figure F.6, determines the values of modules in symbol character 1 (i.e. making the module identified in Figure F.7 as 1.b represent module 1,2). The diagonal line continues through modules 2.8 and 3.8.

f) At this point, the diagonal line crosses the top row boundary. The next diagonal line is started 4 modules to the right in the top row, or in the case of the 8 x 8 mapping matrix, 3 modules right and 1 module down; i.e. the diagonal line is always displaced by 4 modules. Symbol characters are numbered in order, based on the placement path crossing module 8. Thus the next characters are determined by the downward diagonal line crossing modules 4.8, 5.8, 6.8 and so on.

g) As shown in Figure F.8, the placement path continues as diagonal lines four modules to the right (or four modules down, or combinations thereof) from the previous diagonal line. The first, and all odd numbered, diagonal lines map the symbol character sequence from bottom left to top right. The second, and all even numbered, diagonal lines map the symbol character sequence from the top right to the bottom left.

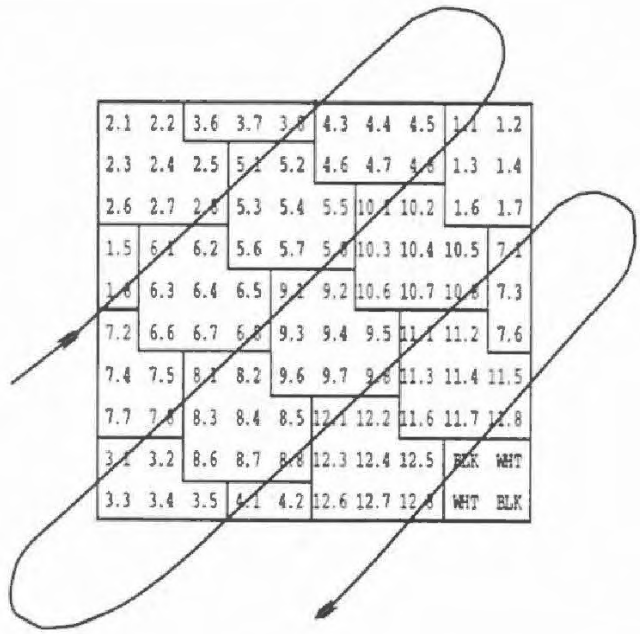


Figure F.8 — Symbol character placement sequence

h) When the placement path encounters a non-standard symbol character shape, which is not completely contained within the boundaries of the mapping matrix, that symbol character is continued on the opposite side of the matrix. This has the effect of numbering the opposite portions of these symbol characters before the placement path crosses that position. For example, in the illustrated mapping matrix (see Figure F.8) the other portions of symbol character 3 and 7 are pre-numbered before the placement path crosses them. Thus the placement path only numbers un-numbered symbol characters. These boundary and corner conditions are specified in Table F.1. This can be seen in Figure F.8 for symbol characters 1, 3, 4, and 7. The corner conditions also affect the numbering sequence. The bottom left corner as illustrated in:

Figure F.3 is numbered immediately before the symbol character above it (see Figures F.11 and F.18 for examples).

Figure F.4 is numbered immediately before the symbol character above it (see Figures F.12 and F.19 for examples).

Figure F.5 is numbered immediately after the symbol character to its right (see Figure F.13 for an example).

Figure F.6 is numbered immediately before the symbol character above it (see Figure F.15 for an example).

The remaining modules of the corner are numbered before the placement path crosses them.

i) The placement procedure continues until all symbol characters are placed, and it ends in the lower right of the mapping matrix. Four sizes of mapping matrix (10 x 10, 14 x 14, 18 x 18, and 22 x 22) have a 2 x 2 area remaining in the bottom right hand corner. The top left and bottom right modules of this area are dark (nominally encoding binary 1). This is illustrated in Figure F.8.

Typical mapping matrices conforming to this procedure are illustrated in F.3. Figures F.9 to F.15 cover respective cases 1 to 7 for boundary placement. Figures F.16 to F.19 are another set of examples for cases 1 to 4. F.1 provides a C language program capable of mapping all encoded bits into the appropriate mapping matrix.

F.3 Symbol character placement examples for ECC 200

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6

Figure F.9 — Codeword placement for square mapping matrix of size 8

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	1.1	1.2
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	1.3	1.4
2.6	2.7	2.8	5.3	5.4	5.5	10.1	10.2	1.6	1.7
1.5	6.1	6.2	5.6	5.7	5.8	10.3	10.4	10.5	7.1
1.8	6.3	6.4	6.5	9.1	9.2	10.6	10.7	10.8	7.3
7.2	6.6	6.7	6.8	9.3	9.4	9.5	11.1	11.2	7.6
7.4	7.5	8.1	8.2	9.6	9.7	9.8	11.3	11.4	11.5
7.7	7.8	8.3	8.4	8.5	12.1	12.2	11.6	11.7	11.8
3.1	3.2	8.6	8.7	8.8	12.3	12.4	12.5	BLK	WHT
3.3	3.4	3.5	4.1	4.2	12.6	12.7	12.8	WHT	BLK

Figure F.10 — Codeword placement for square mapping matrix of size 10

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	13.1	13.2	8.4	8.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	8.6
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	8.7
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	14.1	14.2	8.8
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	14.3	14.4	14.5
7.2	6.6	6.7	6.8	11.3	11.4	11.5	15.1	15.2	14.6	14.7	14.8
7.4	7.5	10.1	10.2	11.6	11.7	11.8	15.3	15.4	15.5	1.1	1.2
7.7	7.8	10.3	10.4	10.5	16.1	16.2	15.6	15.7	15.8	1.3	1.4
9.1	9.2	10.6	10.7	10.8	16.3	16.4	16.5	18.1	18.2	1.6	1.7
9.3	9.4	9.5	17.1	17.2	16.6	16.7	16.8	18.3	18.4	18.5	7.1
9.6	9.7	9.8	17.3	17.4	17.5	3.1	3.2	18.6	18.7	18.8	7.3
8.1	8.2	8.3	17.6	17.7	17.8	3.3	3.4	3.5	4.1	4.2	7.6

Figure F.11 — Codeword placement for square mapping matrix of size 12

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	13.1	13.2	8.4	8.5	8.6	8.7
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	14.1	14.2	8.8
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	14.3	14.4	14.5
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	15.1	15.2	14.6	14.7	14.8
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	15.3	15.4	15.5	1.1	1.2
7.2	6.6	6.7	6.8	11.3	11.4	11.5	16.1	16.2	15.6	15.7	15.8	1.3	1.4
7.4	7.5	10.1	10.2	11.6	11.7	11.8	16.3	16.4	16.5	22.1	22.2	1.6	1.7
7.7	7.8	10.3	10.4	10.5	17.1	17.2	16.6	16.7	16.8	22.3	22.4	22.5	7.1
9.1	9.2	10.6	10.7	10.8	17.3	17.4	17.5	21.1	21.2	22.6	22.7	22.8	7.3
9.3	9.4	9.5	18.1	18.2	17.6	17.7	17.8	21.3	21.4	21.5	23.1	23.2	7.6
9.6	9.7	9.8	18.3	18.4	18.5	20.1	20.2	21.6	21.7	21.8	23.3	23.4	23.5
8.1	19.1	19.2	18.6	18.7	18.8	20.3	20.4	20.5	24.1	24.2	23.6	23.7	23.8
8.2	19.3	19.4	19.5	3.1	3.2	20.6	20.7	20.8	24.3	24.4	24.5	BLK	WHT
8.3	19.6	19.7	19.8	3.3	3.4	3.5	4.1	4.2	24.6	24.7	24.8	WHT	BLK

Figure F.12 — Codeword placement for square mapping matrix of size 14

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	9.1	9.2	10.6	10.7	10.8	7.3	7.4	7.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	9.3	9.4	9.5	11.1	11.2	7.6	7.7	7.8
2.6	2.7	2.8	5.3	5.4	5.5	8.1	8.2	9.6	9.7	9.8	11.3	11.4	11.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	8.3	8.4	8.5	12.1	12.2	11.6	11.7	11.8	1.3	1.4
1.8	6.3	6.4	6.5	3.1	3.2	8.6	8.7	8.8	12.3	12.4	12.5	10.1	10.2	1.6	1.7
7.1	6.6	6.7	6.8	3.3	3.4	3.5	4.1	4.2	12.6	12.7	12.8	10.3	10.4	10.5	7.2

Figure F.13 — Codeword placement for 6 x 16 rectangular mapping matrix

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	11.1	11.2	12.6	12.7	12.8	13.3	13.4	13.5	21.1	21.2	22.6	22.7	22.8	23.3	23.4	23.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	11.3	11.4	11.5	14.1	14.2	13.6	13.7	13.8	21.3	21.4	21.5	24.1	24.2	23.6	23.7	23.8
2.6	2.7	2.8	5.3	5.4	5.5	10.1	10.2	11.6	11.7	11.8	14.3	14.4	14.5	20.1	20.2	21.6	21.7	21.8	24.3	24.4	24.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	10.3	10.4	10.5	15.1	15.2	14.6	14.7	14.8	20.3	20.4	20.5	25.1	25.2	24.6	24.7	24.8	1.3	1.4
1.8	6.3	6.4	6.5	9.1	9.2	10.6	10.7	10.8	15.3	15.4	15.5	19.1	19.2	20.6	20.7	20.8	25.3	25.4	25.5	29.1	29.2	1.6	1.7
7.2	6.6	6.7	6.8	9.3	9.4	9.5	16.1	16.2	15.6	15.7	15.8	19.3	19.4	19.5	26.1	26.2	25.6	25.7	25.8	29.3	29.4	29.5	7.1
7.4	7.5	8.1	8.2	9.6	9.7	9.8	16.3	16.4	16.5	18.1	18.2	19.6	19.7	19.8	26.3	26.4	26.5	28.1	28.2	29.6	29.7	29.8	7.3
7.7	7.8	8.3	8.4	8.5	17.1	17.2	16.6	16.7	16.8	18.3	18.4	18.5	27.1	27.2	26.6	26.7	26.8	28.3	28.4	28.5	30.1	30.2	7.6
3.1	3.2	8.6	8.7	8.8	17.3	17.4	17.5	12.1	12.2	18.6	18.7	18.8	27.3	27.4	27.5	22.1	22.2	28.6	28.7	28.8	30.3	30.4	30.5
3.3	3.4	3.5	4.1	4.2	17.6	17.7	17.8	12.3	12.4	12.5	13.1	13.2	27.6	27.7	27.8	22.3	22.4	22.5	23.1	23.2	30.6	30.7	30.8

Figure F.14 — Codeword placement for 10 x 24 rectangular mapping matrix

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	8.1	8.2	9.6	9.7	9.8	10.3	10.4	10.5	14.1	14.2	15.6	15.7	15.8	16.3	16.4	16.5	20.1	20.2	1.4	1.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	8.3	8.4	8.5	11.1	11.2	10.6	10.7	10.8	14.3	14.4	14.5	17.1	17.2	16.6	16.7	16.8	20.3	20.4	20.5	1.6
2.6	2.7	2.8	5.3	5.4	5.5	7.1	7.2	8.6	8.7	8.8	11.3	11.4	11.5	13.1	13.2	14.6	14.7	14.8	17.3	17.4	17.5	19.1	19.2	20.6	20.7	20.8	1.7
1.1	6.1	6.2	5.6	5.7	5.8	7.3	7.4	7.5	12.1	12.2	11.6	11.7	11.8	13.3	13.4	13.5	18.1	18.2	17.6	17.7	17.8	19.3	19.4	19.5	21.1	21.2	1.8
1.2	6.3	6.4	6.5	3.1	3.2	7.6	7.7	7.8	12.3	12.4	12.5	9.1	9.2	13.6	13.7	13.8	18.3	18.4	18.5	15.1	15.2	19.6	19.7	19.8	21.3	21.4	21.5
1.3	6.6	6.7	6.8	3.3	3.4	3.5	4.1	4.2	12.6	12.7	12.8	9.3	9.4	9.5	10.1	10.2	18.6	18.7	18.8	15.3	15.4	15.5	16.1	16.2	21.6	21.7	21.8

Figure F.15 — Codeword placement for 6 x 28 rectangular mapping matrix

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	13.1	13.2	14.6	14.7	14.8	15.3	15.4	15.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	16.1	16.2	15.6	15.7	15.8
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	16.3	16.4	16.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	17.1	17.2	16.6	16.7	16.8	1.3	1.4
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	17.3	17.4	17.5	27.1	27.2	1.6	1.7
7.2	6.6	6.7	6.8	11.3	11.4	11.5	18.1	18.2	17.6	17.7	17.8	27.3	27.4	27.5	7.1
7.4	7.5	10.1	10.2	11.6	11.7	11.8	18.3	18.4	18.5	26.1	26.2	27.6	27.7	27.8	7.3
7.7	7.8	10.3	10.4	10.5	19.1	19.2	18.6	18.7	18.8	26.3	26.4	26.5	28.1	28.2	7.6
9.1	9.2	10.6	10.7	10.8	19.3	19.4	19.5	25.1	25.2	26.6	26.7	26.8	28.3	28.4	28.5
9.3	9.4	9.5	20.1	20.2	19.6	19.7	19.8	25.3	25.4	25.5	29.1	29.2	28.6	28.7	28.8
9.6	9.7	9.8	20.3	20.4	20.5	24.1	24.2	25.6	25.7	25.8	29.3	29.4	29.5	8.1	8.2
8.5	21.1	21.2	20.6	20.7	20.8	24.3	24.4	24.5	30.1	30.2	29.6	29.7	29.8	8.3	8.4
8.8	21.3	21.4	21.5	23.1	23.2	24.6	24.7	24.8	30.3	30.4	30.5	32.1	32.2	8.6	8.7
22.2	21.6	21.7	21.8	23.3	23.4	23.5	31.1	31.2	30.6	30.7	30.8	32.3	32.4	32.5	22.1
22.4	22.5	3.1	3.2	23.6	23.7	23.8	31.3	31.4	31.5	14.1	14.2	32.6	32.7	32.8	22.3
22.7	22.8	3.3	3.4	3.5	4.1	4.2	31.6	31.7	31.8	14.3	14.4	14.5	15.1	15.2	22.6

Figure F.16 — Codeword placement for square mapping matrix of size 16

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	13.1	13.2	14.6	14.7	14.8	15.3	15.4	15.5	1.1	1.2
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	16.1	16.2	15.6	15.7	15.8	1.3	1.4
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	16.3	16.4	16.5	29.1	29.2	1.6	1.7
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	17.1	17.2	16.6	16.7	16.8	29.3	29.4	29.5	7.1
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	17.3	17.4	17.5	28.1	28.2	29.6	29.7	29.8	7.3
7.2	6.6	6.7	6.8	11.3	11.4	11.5	18.1	18.2	17.6	17.7	17.8	28.3	28.4	28.5	30.1	30.2	7.6
7.4	7.5	10.1	10.2	11.6	11.7	11.8	18.3	18.4	18.5	27.1	27.2	28.6	28.7	28.8	30.3	30.4	30.5
7.7	7.8	10.3	10.4	10.5	19.1	19.2	18.6	18.7	18.8	27.3	27.4	27.5	31.1	31.2	30.6	30.7	30.8
9.1	9.2	10.6	10.7	10.8	19.3	19.4	19.5	26.1	26.2	27.6	27.7	27.8	31.3	31.4	31.5	8.1	8.2
9.3	9.4	9.5	20.1	20.2	19.6	19.7	19.8	26.3	26.4	26.5	32.1	32.2	31.6	31.7	31.8	8.3	8.4
9.6	9.7	9.8	20.3	20.4	20.5	25.1	25.2	26.6	26.7	26.8	32.3	32.4	32.5	38.1	38.2	8.6	8.7
8.5	21.1	21.2	20.6	20.7	20.8	25.3	25.4	25.5	33.1	33.2	32.6	32.7	32.8	38.3	38.4	38.5	22.1
8.8	21.3	21.4	21.5	24.1	24.2	25.6	25.7	25.8	33.3	33.4	33.5	37.1	37.2	38.6	38.7	38.8	22.3
22.2	21.6	21.7	21.8	24.3	24.4	24.5	34.1	34.2	33.6	33.7	33.8	37.3	37.4	37.5	39.1	39.2	22.6
22.4	22.5	23.1	23.2	24.6	24.7	24.8	34.3	34.4	34.5	36.1	36.2	37.6	37.7	37.8	39.3	39.4	39.5
22.7	22.8	23.3	23.4	23.5	35.1	35.2	34.6	34.7	34.8	36.3	36.4	36.5	40.1	40.2	39.6	39.7	39.8
3.1	3.2	23.6	23.7	23.8	35.3	35.4	35.5	14.1	14.2	36.6	36.7	36.8	40.3	40.4	40.5	BLK	WHT
3.3	3.4	3.5	4.1	4.2	35.6	35.7	35.8	14.3	14.4	14.5	15.1	15.2	40.6	40.7	40.8	WHT	BLK

Figure F.17 — Codeword placement for square mapping matrix of size 18

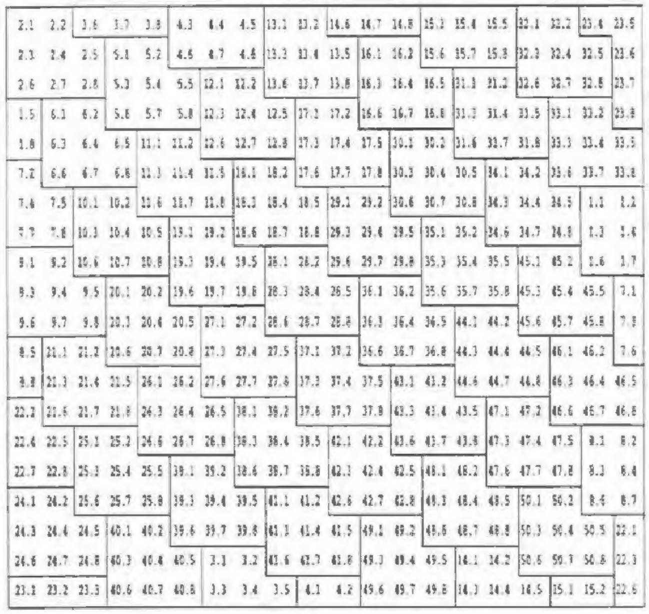


Figure F.18 — Codeword placement for square mapping matrix of size 20

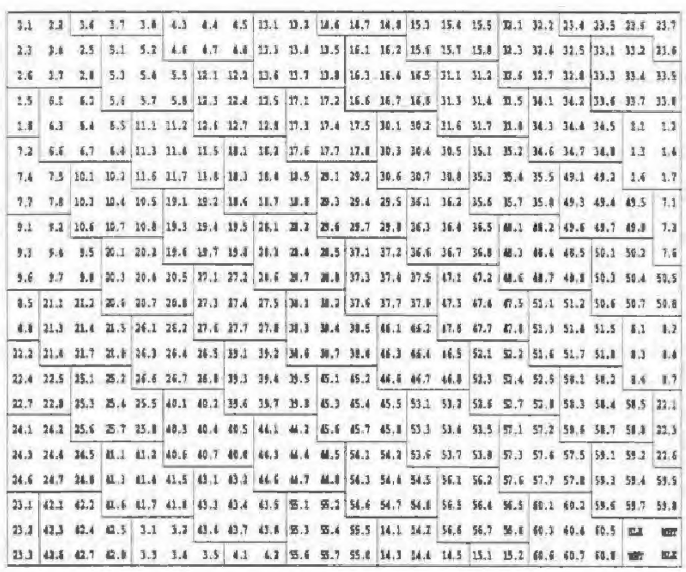


Figure F.19 — Codeword placement for square mapping matrix of size 22

Annex G
(normative)

ECC 000 - 140 symbol attributes

Table G.1 — ECC 000

Symbol size[a]		Data region size		Numeric capacity	Alphanum capacity	8-bit byte capacity	% of codewords used for error correction	% correctable
Row	Col	Row	Col					
9	9	7	7	3	2	1	0,0	0,0
11	11	9	9	12	8	5	0,0	0,0
13	13	11	11	24	16	10	0,0	0,0
15	15	13	13	37	25	16	0,0	0,0
17	17	15	15	53	35	23	0,0	0,0
19	19	17	17	72	48	31	0,0	0,0
21	21	19	19	92	61	40	0,0	0,0
23	23	21	21	115	76	50	0,0	0,0
25	25	23	23	140	93	61	0,0	0,0
27	27	25	25	168	112	73	0,0	0,0
29	29	27	27	197	131	86	0,0	0,0
31	31	29	29	229	153	100	0,0	0,0
33	33	31	31	264	176	115	0,0	0,0
35	35	33	33	300	200	131	0,0	0,0
37	37	35	35	339	226	148	0,0	0,0
39	39	37	37	380	253	166	0,0	0,0
41	41	39	39	424	282	185	0,0	0,0
43	43	41	41	469	313	205	0,0	0,0
45	45	43	43	500	345	226	0,0	0,0
47	47	45	45	560	378	248	0,0	0,0
49	49	47	47	596	413	271	0,0	0,0

[a] excluding quiet zones

Table G.2 — ECC 050

Symbol size[a]		Data region size		Numeric capacity	Alphanum. capacity	8-bit byte capacity	% of codewords used for error correction	% correctable
Row	Col	Row	Col					
11	11	9	9	1	1	0[b]	25,0	2,8
13	13	11	11	10	6	4	25,0	2,8
15	15	13	13	20	13	9	25,0	2,8
17	17	15	15	32	21	14	25,0	2,8
19	19	17	17	46	30	20	25,0	2,8
21	21	19	19	61	41	27	25,0	2,8
23	23	21	21	78	52	34	25,0	2,8
25	25	23	23	97	65	42	25,0	2,8
27	27	25	25	118	78	51	25,0	2,8
29	29	27	27	140	93	61	25,0	2,8
31	31	29	29	164	109	72	25,0	2,8
33	33	31	31	190	126	83	25,0	2,8
35	35	33	33	217	145	95	25,0	2,8
37	37	35	35	246	164	108	25,0	2,8
39	39	37	37	277	185	121	25,0	2,8
41	41	39	39	310	206	135	25,0	2,8
43	43	41	41	344	229	150	25,0	2,8
45	45	43	43	380	253	166	25,0	2,8
47	47	45	45	418	278	183	25,0	2,8
49	49	47	47	457	305	200	25,0	2,8

[a] excluding quiet zone

[b] this combination is not possible

Table G.3 — ECC 080

Symbol size[a]		Data region size		Numeric capacity	Alphanum. capacity	8-bit byte capacity	% of codewords used for error correction	% correctable
Row	Col	Row	Col					
13	13	11	11	4	3	2	33,3	5,5
15	15	13	13	13	9	6	33,3	5,5
17	17	15	15	24	16	10	33,3	5,5
19	19	17	17	36	24	16	33,3	5,5
21	21	19	19	50	33	22	33,3	5,5
23	23	21	21	65	43	28	33,3	5,5
25	25	23	23	82	54	36	33,3	5,5
27	27	25	25	100	67	44	33,3	5,5
29	29	27	27	120	80	52	33,3	5,5
31	31	29	29	141	94	62	33,3	5,5
33	33	31	31	164	109	72	33,3	5,5
35	35	33	33	188	125	82	33,3	5,5
37	37	35	35	214	143	94	33,3	5,5
39	39	37	37	242	161	106	33,3	5,5
41	41	39	39	270	180	118	33,3	5,5
43	43	41	41	301	201	132	33,3	5,5
45	45	43	43	333	222	146	33,3	5,5
47	47	45	45	366	244	160	33,3	5,5
49	49	47	47	402	268	176	33,3	5,5

a excluding quiet zones

Table G.4 — ECC 100

Symbol size[a]		Data region size		Numeric capacity	Alphanum. capacity	8-bit byte capacity	% of codewords used for error correction	% correctable
Row	Col	Row	Col					
13	13	11	11	1	1	0[b]	50,0	12,6
15	15	13	13	8	5	3	50,0	12,6
17	17	15	15	16	11	7	50,0	12,6
19	19	17	17	25	17	11	50,0	12,6
21	21	19	19	36	24	15	50,0	12,6
23	23	21	21	47	31	20	50,0	12,6
25	25	23	23	60	40	26	50,0	12,6
27	27	25	25	73	49	32	50,0	12,6
29	29	27	27	88	59	38	50,0	12,6
31	31	29	29	104	69	45	50,0	12,6
33	33	31	31	121	81	53	50,0	12,6
35	35	33	33	140	93	61	50,0	12,6
37	37	35	35	159	106	69	50,0	12,6
39	39	37	37	180	120	78	50,0	12,6
41	41	39	39	201	134	88	50,0	12,6
43	43	41	41	224	149	98	50,0	12,6
45	45	43	43	248	165	108	50,0	12,6
47	47	45	45	273	182	119	50,0	12,6
49	49	47	47	300	200	131	50,0	12,6

[a] excluding quiet zones

[b] this combination is not possible

Table G.5 — ECC 140

Symbol size[a]		Data region size		Numeric capacity	Alphanum. capacity	8-bit byte capacity	% of codewords used for error correction	% correctable
Row	Col	Row	Col					
17	17	15	15	2	1	1	75,0	25,0
19	19	17	17	6	4	3	75,0	25,0
21	21	19	19	12	8	5	75,0	25,0
23	23	21	21	17	11	7	75,0	25,0
25	25	23	23	24	16	10	75,0	25,0
27	27	25	25	30	20	13	75,0	25,0
29	29	27	27	38	25	16	75,0	25,0
31	31	29	29	46	30	20	75,0	25,0
33	33	31	31	54	36	24	75,0	25,0
35	35	33	33	64	42	28	75,0	25,0
37	37	35	35	73	49	32	75,0	25,0
39	39	37	37	84	56	36	75,0	25,0
41	41	39	39	94	63	41	75,0	25,0
43	43	41	41	106	70	46	75,0	25,0
45	45	43	43	118	78	51	75,0	25,0
47	47	45	45	130	87	57	75,0	25,0
49	49	47	47	144	96	63	75,0	25,0

[a] excluding quiet zones

Annex H
(normative)

ECC 000 - 140 data module placement grids

Table H.1 — 7 x 7 data

2	45	10	38	24	21	1
12	40	26	5	33	19	47
22	31	29	15	43	8	36
34	20	48	13	41	27	6
44	9	37	23	17	30	16
39	25	4	32	18	46	11
0	28	14	42	7	35	3

Table H.2 — 9 x 9 data

2	19	55	10	46	28	64	73	1
62	17	53	35	71	8	80	44	26
49	31	67	4	76	40	22	58	13
69	6	78	42	24	60	15	51	33
74	38	20	56	11	47	29	65	37
25	61	16	52	34	70	7	79	43
12	48	30	66	63	75	39	21	57
32	68	5	77	41	23	59	14	50
0	72	36	18	54	9	45	27	3

Table H.3 — 11 x 11 data

2	26	114	70	15	103	59	37	81	4	1
117	73	18	106	62	40	84	7	95	51	29
12	100	56	34	78	92	89	45	23	111	67
65	43	87	10	98	54	32	120	76	21	109
82	5	93	49	27	115	71	16	104	60	38
96	52	30	118	74	19	107	63	41	85	8
24	112	68	13	101	57	35	79	48	90	46
75	20	108	64	42	86	9	97	53	31	119
102	58	36	80	77	91	47	25	113	69	14
39	83	6	94	50	28	116	72	17	105	61
0	88	44	22	110	66	11	99	55	33	3

Table H.4 — 13 x 13 data

2	159	29	133	81	16	120	68	42	146	94	91	1
37	141	89	24	128	76	50	154	102	11	115	63	167
83	18	122	70	44	148	96	5	109	57	161	31	135
125	73	47	151	99	8	112	60	164	34	138	86	21
40	144	92	107	105	53	157	27	131	79	14	118	66
103	12	116	64	168	38	142	90	25	129	77	51	155
110	58	162	32	136	84	19	123	71	45	149	97	6
165	35	139	87	22	126	74	48	152	100	9	113	61
132	80	15	119	67	41	145	93	55	106	54	158	28
23	127	75	49	153	101	10	114	62	166	36	140	88
69	43	147	95	4	108	56	160	30	134	82	17	121
150	98	7	111	59	163	33	137	85	20	124	72	46
0	104	52	156	26	130	78	13	117	65	39	143	3

Table H.5 — 15 x 15 data

2	187	37	157	97	217	22	142	82	202	52	172	112	7	1
41	161	101	221	26	146	86	206	56	176	116	11	131	71	191
93	213	18	138	78	198	48	168	108	105	123	63	183	33	153
28	148	88	208	58	178	118	13	133	73	193	43	163	103	223
80	200	50	170	110	5	125	65	185	35	155	95	215	20	140
54	174	114	9	129	69	189	39	159	99	219	24	144	84	204
106	127	121	61	181	31	151	91	211	16	136	76	196	46	166
134	74	194	44	164	104	224	29	149	89	209	59	179	119	14
186	36	156	96	216	21	141	81	201	51	171	111	6	126	66
160	100	220	25	145	85	205	55	175	115	10	130	70	190	40
212	17	137	77	197	47	167	107	67	122	62	182	32	152	92
147	87	207	57	177	117	12	132	72	192	42	162	102	222	27
199	49	169	109	4	124	64	184	34	154	94	214	19	139	79
173	113	8	128	68	188	38	158	98	218	23	143	83	203	53
0	120	60	180	30	150	90	210	15	135	75	195	45	165	3

Table H.6 — 17 x 17 data

2	69	205	35	171	103	239	18	154	86	222	52	188	120	256	273	1
220	50	186	118	254	33	169	101	237	67	203	135	271	16	288	152	84
178	110	246	25	161	93	229	59	195	127	263	8	280	144	76	212	42
250	29	165	97	233	63	199	131	267	12	284	148	80	216	46	182	114
157	89	225	55	191	123	259	4	276	140	72	208	38	174	106	242	21
235	65	201	133	269	14	286	150	82	218	48	184	116	252	31	167	99
193	125	261	6	278	142	74	210	40	176	108	244	23	159	91	227	57
265	10	282	146	78	214	44	180	112	248	27	163	95	231	61	197	129
274	138	70	206	36	172	104	240	19	155	87	223	53	189	121	257	137
83	219	49	185	117	253	32	168	100	236	66	202	134	270	15	287	151
41	177	109	245	24	160	92	228	58	194	126	262	7	279	143	75	211
113	249	28	164	96	232	62	198	130	266	11	283	147	79	215	45	181
20	156	88	224	54	190	122	258	255	275	139	71	207	37	173	105	241
98	234	64	200	132	268	13	285	149	81	217	47	183	115	251	30	166
56	192	124	260	5	277	141	73	209	39	175	107	243	22	158	90	226
128	264	9	281	145	77	213	43	179	111	247	26	162	94	230	60	196
0	272	136	68	204	34	170	102	238	17	153	85	221	51	187	119	3

Table H.7 — 19 x 19 data

2	82	234	44	348	196	120	272	25	329	177	101	253	63	215	139	291	6	1
239	49	353	201	125	277	30	334	182	106	258	68	220	144	296	11	315	163	87
343	191	115	267	20	324	172	96	248	58	210	134	286	310	305	153	77	229	39
132	284	37	341	189	113	265	75	227	151	303	18	322	170	94	246	56	360	208
28	332	180	104	256	66	218	142	294	9	313	161	85	237	47	351	199	123	275
185	109	261	71	223	147	299	14	318	166	90	242	52	356	204	128	280	33	337
251	61	213	137	289	4	308	156	80	232	42	346	194	118	270	23	327	175	99
225	149	301	16	320	168	92	244	54	358	206	130	282	35	339	187	111	263	73
292	7	311	159	83	235	45	349	197	121	273	26	330	178	102	254	64	216	140
316	164	88	240	50	354	202	126	278	31	335	183	107	259	69	221	145	297	12
78	230	40	344	192	116	268	21	325	173	97	249	59	211	135	287	158	306	154
55	359	207	131	283	36	340	188	112	264	74	226	150	302	17	321	169	93	245
198	122	274	27	331	179	103	255	65	217	141	293	8	312	160	84	236	46	350
279	32	336	184	108	260	70	222	146	298	13	317	165	89	241	51	355	203	127
326	174	98	250	60	212	136	288	285	307	155	79	231	41	345	193	117	269	22
110	262	72	224	148	300	15	319	167	91	243	53	357	205	129	281	34	338	186
62	214	138	290	5	309	157	81	233	43	347	195	119	271	24	328	176	100	252
143	295	10	314	162	86	238	48	352	200	124	276	29	333	181	105	257	67	219
0	304	152	76	228	38	342	190	114	266	19	323	171	95	247	57	209	133	3

Table H.8 — 21 x 21 data

2	88	424	256	46	382	214	130	298	25	361	193	109	277	67	403	235	151	319	4	1
437	269	59	395	227	143	311	38	374	206	122	290	80	416	248	164	332	17	353	185	101
49	385	217	133	301	28	364	196	112	280	70	406	238	154	322	7	343	175	91	427	259
222	138	306	33	369	201	117	285	75	411	243	159	327	12	348	180	96	432	264	54	390
295	22	358	190	106	274	64	400	232	148	316	340	337	169	85	421	253	43	379	211	127
377	209	125	293	83	419	251	167	335	20	356	188	104	440	272	62	398	230	146	314	41
115	283	73	409	241	157	325	10	346	178	94	430	262	52	388	220	136	304	31	367	199
78	414	246	162	330	15	351	183	99	435	267	57	393	225	141	309	36	372	204	120	288
236	152	320	5	341	173	89	425	257	47	383	215	131	299	26	362	194	110	278	68	404
333	18	354	186	102	438	270	60	396	228	144	312	39	375	207	123	291	81	417	249	165
344	176	92	428	260	50	386	218	134	302	29	365	197	113	281	71	407	239	155	323	8
97	433	265	55	391	223	139	307	34	370	202	118	286	76	412	244	160	328	13	349	181
254	44	380	212	128	296	23	359	191	107	275	65	401	233	149	317	172	338	170	86	422
397	229	145	313	40	376	208	124	292	82	418	250	166	334	19	355	187	103	439	271	61
135	303	30	366	198	114	282	72	408	240	156	324	9	345	177	93	429	261	51	387	219
35	371	203	119	287	77	413	245	161	329	14	350	182	98	434	266	56	392	224	140	308
192	108	276	66	402	234	150	318	315	339	171	87	423	255	45	381	213	129	297	24	360
289	79	415	247	163	331	16	352	184	100	436	268	58	394	226	142	310	37	373	205	121
405	237	153	321	6	342	174	90	426	258	48	384	216	132	300	27	363	195	111	279	69
158	326	11	347	179	95	431	263	53	389	221	137	305	32	368	200	116	284	74	410	242
0	336	168	84	420	252	42	378	210	126	294	21	357	189	105	273	63	399	231	147	3

Table H.9 — 23 x 23 data

2	102	470	286	56	424	240	148	516	332	33	401	217	125	493	309	79	447	263	171	355	10	1
476	292	62	430	246	154	522	338	39	407	223	131	499	315	85	453	269	177	361	16	384	200	108
50	418	234	142	510	326	27	395	211	119	487	303	73	441	257	165	349	4	372	188	96	464	280
249	157	525	341	42	410	226	134	502	318	88	456	272	180	364	19	387	203	111	479	295	65	433
513	329	30	398	214	122	490	306	76	444	260	168	352	7	375	191	99	467	283	53	421	237	145
36	404	220	128	496	312	82	450	266	174	358	13	381	197	105	473	289	59	427	243	151	519	335
208	116	484	300	70	438	254	162	346	378	369	185	93	461	277	47	415	231	139	507	323	24	392
505	321	91	459	275	183	367	22	390	206	114	482	298	68	436	252	160	528	344	45	413	229	137
80	448	264	172	356	11	379	195	103	471	287	57	425	241	149	517	333	34	402	218	126	494	310
270	178	362	17	385	201	109	477	293	63	431	247	155	523	339	40	408	224	132	500	316	86	454
350	5	373	189	97	465	281	51	419	235	143	511	327	28	396	212	120	488	304	74	442	258	166
388	204	112	480	296	66	434	250	158	526	342	43	411	227	135	503	319	89	457	273	181	365	20
100	468	284	54	422	238	146	514	330	31	399	215	123	491	307	77	445	261	169	353	8	376	192
290	60	428	244	152	520	336	37	405	221	129	497	313	83	451	267	175	359	14	382	198	106	474
416	232	140	508	324	25	393	209	117	485	301	71	439	255	163	347	194	370	186	94	462	278	48
159	527	343	44	412	228	136	504	320	90	458	274	182	366	21	389	205	113	481	297	67	435	251
331	32	400	216	124	492	308	78	446	262	170	354	9	377	193	101	469	285	55	423	239	147	515
406	222	130	498	314	84	452	268	176	360	15	383	199	107	475	291	61	429	245	153	521	337	38
118	486	302	72	440	256	164	348	345	371	187	95	463	279	49	417	233	141	509	325	26	394	210
317	87	455	271	179	363	18	386	202	110	478	294	64	432	248	156	524	340	41	409	225	133	501
443	259	167	351	6	374	190	98	466	282	52	420	236	144	512	328	29	397	213	121	489	305	75
173	357	12	380	196	104	472	288	58	426	242	150	518	334	35	403	219	127	495	311	81	449	265
0	368	184	92	460	276	46	414	230	138	506	322	23	391	207	115	483	299	69	437	253	161	3

Table H.10 — 25 x 25 data

2	603	103	503	303	53	453	253	153	553	353	28	428	228	128	528	328	78	478	278	178	578	378	375	1
123	523	323	73	473	273	173	573	373	48	448	248	148	548	348	98	498	298	198	598	398	23	423	223	623
311	61	461	261	161	561	361	36	436	236	136	536	336	86	486	286	186	586	386	11	411	211	611	111	511
467	267	167	567	367	42	442	242	142	542	342	92	492	292	192	592	392	17	417	217	617	117	517	317	67
155	555	355	30	430	230	130	530	330	80	480	280	180	580	380	5	405	205	605	105	505	305	55	455	255
370	45	445	245	145	545	345	95	495	295	195	595	395	20	420	220	620	120	520	320	70	470	270	170	570
433	233	133	533	333	83	483	283	183	583	383	8	408	208	608	108	508	308	58	458	258	158	558	358	33
139	539	339	89	489	289	189	589	389	14	414	214	614	114	514	314	64	464	264	164	564	364	39	439	239
326	76	476	276	176	576	376	403	401	201	601	101	501	301	51	451	251	151	551	351	26	426	226	126	526
499	299	199	599	399	24	424	224	624	124	524	324	74	474	274	174	574	374	49	449	249	149	549	349	99
187	587	387	12	412	212	612	112	512	312	62	462	262	162	562	362	37	437	237	137	537	337	87	487	287
393	18	418	218	618	118	518	318	68	468	268	168	568	368	43	443	243	143	543	343	93	493	293	193	593
406	206	606	106	506	306	56	456	256	156	556	356	31	431	231	131	531	331	81	481	281	181	581	381	6
621	121	521	321	71	471	271	171	571	371	46	446	246	146	546	346	96	496	296	196	596	396	21	421	221
509	309	59	459	259	159	559	359	34	434	234	134	534	334	84	484	284	184	584	384	9	409	209	609	109
65	465	265	165	565	365	40	440	240	140	540	340	90	490	290	190	590	390	15	415	215	615	115	515	315
252	152	552	352	27	427	227	127	527	327	77	477	277	177	577	377	203	402	202	602	102	502	302	52	452
572	372	47	447	247	147	547	347	97	497	297	197	597	397	22	422	222	622	122	522	322	72	472	272	172
35	435	235	135	535	335	85	485	285	185	585	385	10	410	210	610	110	510	310	60	460	260	160	560	360
241	141	541	341	91	491	291	191	591	391	16	416	216	616	116	516	316	66	466	266	166	566	366	41	441
529	329	79	479	279	179	579	379	4	404	204	604	104	504	304	54	454	254	154	554	354	29	429	229	129
94	494	294	194	594	394	19	419	219	619	119	519	319	69	469	269	169	569	369	44	444	244	144	544	344
282	182	582	382	7	407	207	607	107	507	307	57	457	257	157	557	357	32	432	232	132	532	332	82	482
588	388	13	413	213	613	113	513	313	63	463	263	163	563	363	38	438	238	138	538	338	88	488	288	188
0	400	200	600	100	500	300	50	450	250	150	550	350	25	425	225	125	525	325	75	475	275	175	575	3

Table H.11 — 27 x 27 data

2	658	118	550	334	64	496	280	712	172	604	388	37	469	253	685	145	577	361	91	523	307	199	631	415	10	1
125	557	341	71	503	287	719	179	611	395	44	476	260	692	152	584	368	98	530	314	206	638	422	17	449	233	665
327	57	489	273	705	165	597	381	30	462	246	678	138	570	354	84	516	300	192	624	408	405	435	219	651	111	543
511	295	727	187	619	403	52	484	268	700	160	592	376	106	538	322	214	646	430	25	457	241	673	133	565	349	79
714	174	606	390	39	471	255	687	147	579	363	93	525	309	201	633	417	12	444	228	660	120	552	336	66	498	282
613	397	46	478	262	694	154	586	370	100	532	316	208	640	424	19	451	235	667	127	559	343	73	505	289	721	181
32	464	248	680	140	572	356	86	518	302	194	626	410	5	437	221	653	113	545	329	59	491	275	707	167	599	383
265	697	157	589	373	103	535	319	211	643	427	22	454	238	670	130	562	346	76	508	292	724	184	616	400	49	481
143	575	359	89	521	305	197	629	413	8	440	224	656	116	548	332	62	494	278	710	170	602	386	35	467	251	683
366	96	528	312	204	636	420	15	447	231	663	123	555	339	69	501	285	717	177	609	393	42	474	258	690	150	582
514	298	190	622	406	442	433	217	649	109	541	325	55	487	271	703	163	595	379	28	460	244	676	136	568	352	82
215	647	431	26	458	242	674	134	566	350	80	512	296	728	188	620	404	53	485	269	701	161	593	377	107	539	323
418	13	445	229	661	121	553	337	67	499	283	715	175	607	391	40	472	256	688	148	580	364	94	526	310	202	634
452	236	668	128	560	344	74	506	290	722	182	614	398	47	479	263	695	155	587	371	101	533	317	209	641	425	20
654	114	546	330	60	492	276	708	168	600	384	33	465	249	681	141	573	357	87	519	303	195	627	411	6	438	222
563	347	77	509	293	725	185	617	401	50	482	266	698	158	590	374	104	536	320	212	644	428	23	455	239	671	131
63	495	279	711	171	603	387	36	468	252	684	144	576	360	90	522	306	198	630	414	9	441	225	657	117	549	333
286	718	178	610	394	43	475	259	691	151	583	367	97	529	313	205	637	421	16	448	232	664	124	556	340	70	502
164	596	380	29	461	245	677	137	569	353	83	515	299	191	623	407	226	434	218	650	110	542	326	56	488	272	704
402	51	483	267	699	159	591	375	105	537	321	213	645	429	24	456	240	672	132	564	348	78	510	294	726	186	618
470	254	686	146	578	362	92	524	308	200	632	416	11	443	227	659	119	551	335	65	497	281	713	173	605	389	38
693	153	585	369	99	531	315	207	639	423	18	450	234	666	126	558	342	72	504	288	720	180	612	396	45	477	261
571	355	85	517	301	193	625	409	4	436	220	652	112	544	328	58	490	274	706	166	598	382	31	463	247	679	139
102	534	318	210	642	426	21	453	237	669	129	561	345	75	507	291	723	183	615	399	48	480	264	696	156	588	372
304	196	628	412	7	439	223	655	115	547	331	61	493	277	709	169	601	385	34	466	250	682	142	574	358	88	520
635	419	14	446	230	662	122	554	338	68	500	284	716	176	608	392	41	473	257	689	149	581	365	95	527	311	203
0	432	216	648	108	540	324	54	486	270	702	162	594	378	27	459	243	675	135	567	351	81	513	297	189	621	3

Table H.12 — 29 x 29 data

2	703	123	587	355	819	65	529	297	761	181	645	413	36	500	268	732	152	616	384	94	558	326	790	210	674	442	7	1
141	605	373	837	83	547	315	779	199	663	431	54	518	286	750	170	634	402	112	576	344	808	228	692	460	25	489	257	721
359	823	69	533	301	765	185	649	417	40	504	272	736	156	620	388	98	562	330	794	214	678	446	11	475	243	707	127	591
76	540	308	772	192	656	424	47	511	279	743	163	627	395	105	569	337	801	221	685	453	18	482	250	714	134	598	366	830
293	757	177	641	409	32	496	264	728	148	612	380	90	554	322	786	206	670	438	435	467	235	699	119	583	351	815	61	525
201	665	433	56	520	288	752	172	636	404	114	578	346	810	230	694	462	27	491	259	723	143	607	375	839	85	549	317	781
419	42	506	274	738	158	622	390	100	564	332	796	216	680	448	13	477	245	709	129	593	361	825	71	535	303	767	187	651
513	281	745	165	629	397	107	571	339	803	223	687	455	20	484	252	716	136	600	368	832	78	542	310	774	194	658	426	49
730	150	614	382	92	556	324	788	208	672	440	5	469	237	701	121	585	353	817	63	527	295	759	179	643	411	34	498	266
632	400	110	574	342	806	226	690	458	23	487	255	719	139	603	371	835	81	545	313	777	197	661	429	52	516	284	748	168
96	560	328	792	212	676	444	9	473	241	705	125	589	357	821	67	531	299	763	183	647	415	38	502	270	734	154	618	386
335	799	219	683	451	16	480	248	712	132	596	364	828	74	538	306	770	190	654	422	45	509	277	741	161	625	393	103	567
204	668	436	471	465	233	697	117	581	349	813	59	523	291	755	175	639	407	30	494	262	726	146	610	378	88	552	320	784
463	28	492	260	724	144	608	376	840	86	550	318	782	202	666	434	57	521	289	753	173	637	405	115	579	347	811	231	695
478	246	710	130	594	362	826	72	536	304	768	188	652	420	43	507	275	739	159	623	391	101	565	333	797	217	681	449	14
717	137	601	369	833	79	543	311	775	195	659	427	50	514	282	746	166	630	398	108	572	340	804	224	688	456	21	485	253
586	354	818	64	528	296	760	180	644	412	35	499	267	731	151	615	383	93	557	325	789	209	673	441	6	470	238	702	122
836	82	546	314	778	198	662	430	53	517	285	749	169	633	401	111	575	343	807	227	691	459	24	488	256	720	140	604	372
532	300	764	184	648	416	39	503	271	735	155	619	387	97	561	329	793	213	677	445	10	474	242	706	126	590	358	822	68
771	191	655	423	46	510	278	742	162	626	394	104	568	336	800	220	684	452	17	481	249	713	133	597	365	829	75	539	307
640	408	31	495	263	727	147	611	379	89	553	321	785	205	669	437	239	466	234	698	118	582	350	814	60	524	292	756	176
55	519	287	751	171	635	403	113	577	345	809	229	693	461	26	490	258	722	142	606	374	838	84	548	316	780	200	664	432
273	737	157	621	389	99	563	331	795	215	679	447	12	476	244	708	128	592	360	824	70	534	302	766	186	650	418	41	505
164	628	396	106	570	338	802	222	686	454	19	483	251	715	135	599	367	831	77	541	309	773	193	657	425	48	512	280	744
381	91	555	323	787	207	671	439	4	468	236	700	120	584	352	816	62	526	294	758	178	642	410	33	497	265	729	149	613
573	341	805	225	689	457	22	486	254	718	138	602	370	834	80	544	312	776	196	660	428	51	515	283	747	167	631	399	109
791	211	675	443	8	472	240	704	124	588	356	820	66	530	298	762	182	646	414	37	501	269	733	153	617	385	95	559	327
682	450	15	479	247	711	131	595	363	827	73	537	305	769	189	653	421	44	508	276	740	160	624	392	102	566	334	798	218
0	464	232	696	116	580	348	812	58	522	290	754	174	638	406	29	493	261	725	145	609	377	87	551	319	783	203	667	3

Table H.13 — 31 x 31 data

2	759	139	635	387	883	77	573	325	821	201	697	449	945	46	542	294	790	170	666	418	914	108	604	356	852	232	728	480	15	1
147	643	395	891	85	581	333	829	209	705	457	953	54	550	302	798	178	674	426	922	116	612	364	860	240	736	488	23	519	271	767
379	875	69	565	317	813	193	689	441	937	38	534	286	782	162	658	410	906	100	596	348	844	224	720	472	7	503	255	751	131	627
89	585	337	833	213	709	461	957	58	554	306	802	182	678	430	926	120	616	368	864	244	740	492	27	523	275	771	151	647	399	895
321	817	197	693	445	941	42	538	290	786	166	662	414	910	104	600	352	848	228	724	476	11	507	259	755	135	631	383	879	73	569
205	701	453	949	50	546	298	794	174	670	422	918	112	608	360	856	236	732	484	19	515	267	763	143	639	391	887	81	577	329	825
437	933	34	530	282	778	158	654	406	902	96	592	344	840	220	716	468	465	499	251	747	127	623	375	871	65	561	313	809	189	685
60	556	308	804	184	680	432	928	122	618	370	866	246	742	494	29	525	277	773	153	649	401	897	91	587	339	835	215	711	463	959
292	788	168	664	416	912	106	602	354	850	230	726	478	13	509	261	757	137	633	385	881	75	571	323	819	199	695	447	943	44	540
176	672	424	920	114	610	362	858	238	734	486	21	517	269	765	145	641	393	889	83	579	331	827	207	703	455	951	52	548	300	796
408	904	98	594	346	842	222	718	470	5	501	253	749	129	625	377	873	67	563	315	811	191	687	439	935	36	532	284	780	160	656
118	614	366	862	242	738	490	25	521	273	769	149	645	397	893	87	583	335	831	211	707	459	955	56	552	304	800	180	676	428	924
350	846	226	722	474	9	505	257	753	133	629	381	877	71	567	319	815	195	691	443	939	40	536	288	784	164	660	412	908	102	598
234	730	482	17	513	265	761	141	637	389	885	79	575	327	823	203	699	451	947	48	544	296	792	172	668	420	916	110	606	358	854
466	511	497	249	745	125	621	373	869	63	559	311	807	187	683	435	931	32	528	280	776	156	652	404	900	94	590	342	838	218	714
526	278	774	154	650	402	898	92	588	340	836	216	712	464	960	61	557	309	805	185	681	433	929	123	619	371	867	247	743	495	30
758	138	634	386	882	76	572	324	820	200	696	448	944	45	541	293	789	169	665	417	913	107	603	355	851	231	727	479	14	510	262
642	394	890	84	580	332	828	208	704	456	952	53	549	301	797	177	673	425	921	115	611	363	859	239	735	487	22	518	270	766	146
874	68	564	316	812	192	688	440	936	37	533	285	781	161	657	409	905	99	595	347	843	223	719	471	6	502	254	750	130	626	378
584	336	832	212	708	460	956	57	553	305	801	181	677	429	925	119	615	367	863	243	739	491	26	522	274	770	150	646	398	894	88
816	196	692	444	940	41	537	289	785	165	661	413	909	103	599	351	847	227	723	475	10	506	258	754	134	630	382	878	72	568	320
700	452	948	49	545	297	793	173	669	421	917	111	607	359	855	235	731	483	18	514	266	762	142	638	390	886	80	576	328	824	204
932	33	529	281	777	157	653	405	901	95	591	343	839	219	715	467	263	498	250	746	126	622	374	870	64	560	312	808	188	684	436
555	307	803	183	679	431	927	121	617	369	865	245	741	493	28	524	276	772	152	648	400	896	90	586	338	834	214	710	462	958	59
787	167	663	415	911	105	601	353	849	229	725	477	12	508	260	756	136	632	384	880	74	570	322	818	198	694	446	942	43	539	291
671	423	919	113	609	361	857	237	733	485	20	516	268	764	144	640	392	888	82	578	330	826	206	702	454	950	51	547	299	795	175
903	97	593	345	841	221	717	469	4	500	252	748	128	624	376	872	66	562	314	810	190	686	438	934	35	531	283	779	159	655	407
613	365	861	241	737	489	24	520	272	768	148	644	396	892	86	582	334	830	210	706	458	954	55	551	303	799	179	675	427	923	117
845	225	721	473	8	504	256	752	132	628	380	876	70	566	318	814	194	690	442	938	39	535	287	783	163	659	411	907	101	597	349
729	481	16	512	264	760	140	636	388	884	78	574	326	822	202	698	450	946	47	543	295	791	171	667	419	915	109	605	357	853	233
0	496	248	744	124	620	372	868	62	558	310	806	186	682	434	930	31	527	279	775	155	651	403	899	93	589	341	837	217	713	3

Table H.14 — 33 x 33 data

2	265	793	133	661	397	925	67	595	331	859	199	727	463	991	34	562	298	826	166	694	430	958	100	628	364	892	232	760	496	1024	1057	1
824	164	692	428	956	98	626	362	890	230	758	494	1022	65	593	329	857	197	725	461	989	131	659	395	923	263	791	527	1055	32	1088	560	296
676	412	940	82	610	346	874	214	742	478	1006	49	577	313	841	181	709	445	973	115	643	379	907	247	775	511	1039	16	1072	544	280	808	148
948	90	618	354	882	222	750	486	1014	57	585	321	849	189	717	453	981	123	651	387	915	255	783	519	1047	24	1080	552	288	816	156	684	420
602	338	866	206	734	470	998	41	569	305	833	173	701	437	965	107	635	371	899	239	767	503	1031	8	1064	536	272	800	140	668	404	932	74
886	226	754	490	1018	61	589	325	853	193	721	457	985	127	655	391	919	259	787	523	1051	28	1084	556	292	820	160	688	424	952	94	622	358
738	474	1002	45	573	309	837	177	705	441	969	111	639	375	903	243	771	507	1035	12	1068	540	276	804	144	672	408	936	78	606	342	870	210
1010	53	581	317	845	185	713	449	977	119	647	383	911	251	779	515	1043	20	1076	548	284	812	152	680	416	944	86	614	350	878	218	746	482
565	301	829	169	697	433	961	103	631	367	895	235	763	499	1027	4	1060	532	268	796	136	664	400	928	70	598	334	862	202	730	466	994	37
855	195	723	459	987	129	657	393	921	261	789	525	1053	30	1086	558	294	822	162	690	426	954	96	624	360	888	228	756	492	1020	63	591	327
707	443	971	113	641	377	905	245	773	509	1037	14	1070	542	278	806	146	674	410	938	80	608	344	872	212	740	476	1004	47	575	311	839	179
979	121	649	385	913	253	781	517	1045	22	1078	550	286	814	154	682	418	946	88	616	352	880	220	748	484	1012	55	583	319	847	187	715	451
633	369	897	237	765	501	1029	6	1062	534	270	798	138	666	402	930	72	600	336	864	204	732	468	996	39	567	303	831	171	699	435	963	105
917	257	785	521	1049	26	1082	554	290	818	158	686	422	950	92	620	356	884	224	752	488	1016	59	587	323	851	191	719	455	983	125	653	389
769	505	1033	10	1066	538	274	802	142	670	406	934	76	604	340	868	208	736	472	1000	43	571	307	835	175	703	439	967	109	637	373	901	241
1041	18	1074	546	282	810	150	678	414	942	84	612	348	876	216	744	480	1008	51	579	315	843	183	711	447	975	117	645	381	909	249	777	513
1058	530	266	794	134	662	398	926	68	596	332	860	200	728	464	992	35	563	299	827	167	695	431	959	101	629	365	893	233	761	497	1025	529
295	823	163	691	427	955	97	625	361	889	229	757	493	1021	64	592	328	856	196	724	460	988	130	658	394	922	262	790	526	1054	31	1087	559
147	675	411	939	81	609	345	873	213	741	477	1005	48	576	312	840	180	708	444	972	114	642	378	906	246	774	510	1038	15	1071	543	279	807
419	947	89	617	353	881	221	749	485	1013	56	584	320	848	188	716	452	980	122	650	386	914	254	782	518	1046	23	1079	551	287	815	155	683
73	601	337	865	205	733	469	997	40	568	304	832	172	700	436	964	106	634	370	898	238	766	502	1030	7	1063	535	271	799	139	667	403	931
357	885	225	753	489	1017	60	588	324	852	192	720	456	984	126	654	390	918	258	786	522	1050	27	1083	555	291	819	159	687	423	951	93	621
209	737	473	1001	44	572	308	836	176	704	440	968	110	638	374	902	242	770	506	1034	11	1067	539	275	803	143	671	407	935	77	605	341	869
481	1009	52	580	316	844	184	712	448	976	118	646	382	910	250	778	514	1042	19	1075	547	283	811	151	679	415	943	85	613	349	877	217	745
36	564	300	828	168	696	432	960	102	630	366	894	234	762	498	1026	1023	1059	531	267	795	135	663	399	927	69	597	333	861	201	729	465	993
326	854	194	722	458	986	128	656	392	920	260	788	524	1052	29	1085	557	293	821	161	689	425	953	95	623	359	887	227	755	491	1019	62	590
178	706	442	970	112	640	376	904	244	772	508	1036	13	1069	541	277	805	145	673	409	937	79	607	343	871	211	739	475	1003	46	574	310	838
450	978	120	648	384	912	252	780	516	1044	21	1077	549	285	813	153	681	417	945	87	615	351	879	219	747	483	1011	54	582	318	846	186	714
104	632	368	896	236	764	500	1028	5	1061	533	269	797	137	665	401	929	71	599	335	863	203	731	467	995	38	566	302	830	170	698	434	962
388	916	256	784	520	1048	25	1081	553	289	817	157	685	421	949	91	619	355	883	223	751	487	1015	58	586	322	850	190	718	454	982	124	652
240	768	504	1032	9	1065	537	273	801	141	669	405	933	75	603	339	867	207	735	471	999	42	570	306	834	174	702	438	966	108	636	372	900
512	1040	17	1073	545	281	809	149	677	413	941	83	611	347	875	215	743	479	1007	50	578	314	842	182	710	446	974	116	644	380	908	248	776
0	1056	528	264	792	132	660	396	924	66	594	330	858	198	726	462	990	33	561	297	825	165	693	429	957	99	627	363	891	231	759	495	3

Table H.15 — 35 x 35 data

2	290	850	150	710	430	990	80	1200	640	360	920	220	780	500	1060	45	1165	605	325	885	185	745	465	1025	115	675	395	955	255	815	535	1095	10	1
859	159	719	439	999	89	1209	649	369	929	229	789	509	1069	54	1174	614	334	894	194	754	474	1034	124	684	404	964	264	824	544	1104	19	1139	579	299
701	421	981	71	1191	631	351	911	211	771	491	1051	36	1156	596	316	876	176	736	456	1016	106	666	386	946	246	806	526	1086	1130	1121	561	281	841	141
1014	104	1224	664	384	944	244	804	524	1084	69	1189	629	349	909	209	769	489	1049	139	699	419	979	279	839	559	1119	34	1154	594	314	874	174	734	454
1207	647	367	927	227	787	507	1067	52	1172	612	332	892	192	752	472	1032	122	682	402	962	262	822	542	1102	17	1137	577	297	857	157	717	437	997	87
376	936	236	796	516	1076	61	1181	621	341	901	201	761	481	1041	131	691	411	971	271	831	551	1111	26	1146	586	306	866	166	726	446	1006	96	1216	656
218	778	498	1058	43	1163	603	323	883	183	743	463	1023	113	673	393	953	253	813	533	1093	8	1128	568	288	848	148	708	428	988	78	1198	638	358	918
520	1080	65	1185	625	345	905	205	765	485	1045	135	695	415	975	275	835	555	1115	30	1150	590	310	870	170	730	450	1010	100	1220	660	380	940	240	800
48	1168	608	328	888	188	748	468	1028	118	678	398	958	258	818	538	1098	13	1133	573	293	853	153	713	433	993	83	1203	643	363	923	223	783	503	1063
617	337	897	197	757	477	1037	127	687	407	967	267	827	547	1107	22	1142	582	302	862	162	722	442	1002	92	1212	652	372	932	232	792	512	1072	57	1177
879	179	739	459	1019	109	669	389	949	249	809	529	1089	4	1124	564	284	844	144	704	424	984	74	1194	634	354	914	214	774	494	1054	39	1159	599	319
767	487	1047	137	697	417	977	277	837	557	1117	32	1152	592	312	872	172	732	452	1012	102	1222	662	382	942	242	802	522	1082	67	1187	627	347	907	207
1030	120	680	400	960	260	820	540	1100	15	1135	575	295	855	155	715	435	995	85	1205	645	365	925	225	785	505	1065	50	1170	610	330	890	190	750	470
689	409	969	269	829	549	1109	24	1144	584	304	864	164	724	444	1004	94	1214	654	374	934	234	794	514	1074	59	1179	619	339	899	199	759	479	1039	129
951	251	811	531	1091	6	1126	566	286	846	146	706	426	986	76	1196	636	356	916	216	776	496	1056	41	1161	601	321	881	181	741	461	1021	111	671	391
833	553	1113	28	1148	588	308	868	168	728	448	1008	98	1218	658	378	938	238	798	518	1078	63	1183	623	343	903	203	763	483	1043	133	693	413	973	273
1096	11	1131	571	291	851	151	711	431	991	81	1201	641	361	921	221	781	501	1061	46	1166	606	326	886	186	746	466	1026	116	676	396	956	256	816	536
1140	580	300	860	160	720	440	1000	90	1210	650	370	930	230	790	510	1070	55	1175	615	335	895	195	755	475	1035	125	685	405	965	265	825	545	1105	20
282	842	142	702	422	982	72	1192	632	352	912	212	772	492	1052	37	1157	597	317	877	177	737	457	1017	107	667	387	947	247	807	527	1087	570	1122	562
173	733	453	1013	103	1223	663	383	943	243	803	523	1083	68	1188	628	348	908	208	768	488	1048	138	698	418	978	278	838	558	1118	33	1153	593	313	873
436	996	86	1206	646	366	926	226	786	506	1066	51	1171	611	331	891	191	751	471	1031	121	681	401	961	261	821	541	1101	16	1136	576	296	856	156	716
95	1215	655	375	935	235	795	515	1075	60	1180	620	340	900	200	760	480	1040	130	690	410	970	270	830	550	1110	25	1145	585	305	865	165	725	445	1005
637	357	917	217	777	497	1057	42	1162	602	322	882	182	742	462	1022	112	672	392	952	252	812	532	1092	7	1127	567	287	847	147	707	427	987	77	1197
939	239	799	519	1079	64	1184	624	344	904	204	764	484	1044	134	694	414	974	274	834	554	1114	29	1149	589	309	869	169	729	449	1009	99	1219	659	379
782	502	1062	47	1167	607	327	887	187	747	467	1027	117	677	397	957	257	817	537	1097	12	1132	572	292	852	152	712	432	992	82	1202	642	362	922	222
1071	56	1176	616	336	896	196	756	476	1036	126	686	406	966	266	826	546	1106	21	1141	581	301	861	161	721	441	1001	91	1211	651	371	931	231	791	511
1158	598	318	878	178	738	458	1018	108	668	388	948	248	808	528	1088	1085	1123	563	283	843	143	703	423	983	73	1193	633	353	913	213	773	493	1053	38
346	906	206	766	486	1046	136	696	416	976	276	836	556	1116	31	1151	591	311	871	171	731	451	1011	101	1221	661	381	941	241	801	521	1081	66	1186	626
189	749	469	1029	119	679	399	959	259	819	539	1099	14	1134	574	294	854	154	714	434	994	84	1204	644	364	924	224	784	504	1064	49	1169	609	329	889
478	1038	128	688	408	968	268	828	548	1108	23	1143	583	303	863	163	723	443	1003	93	1213	653	373	933	233	793	513	1073	58	1178	618	338	898	198	758
110	670	390	950	250	810	530	1090	5	1125	565	285	845	145	705	425	985	75	1195	635	355	915	215	775	495	1055	40	1160	600	320	880	180	740	460	1020
412	972	272	832	552	1112	27	1147	587	307	867	167	727	447	1007	97	1217	657	377	937	237	797	517	1077	62	1182	622	342	902	202	762	482	1042	132	692
254	814	534	1094	9	1129	569	289	849	149	709	429	989	79	1199	639	359	919	219	779	499	1059	44	1164	604	324	884	184	744	464	1024	114	674	394	954
543	1103	18	1138	578	298	858	158	718	438	998	88	1208	648	368	928	228	788	508	1068	53	1173	613	333	893	193	753	473	1033	123	683	403	963	263	823
0	1120	560	280	840	140	700	420	980	70	1190	630	350	910	210	770	490	1050	35	1155	595	315	875	175	735	455	1015	105	665	385	945	245	805	525	3

Table H.16 — 37 x 37 data

2	302	894	154	1338	746	450	1042	80	1264	672	376	968	228	820	524	1116	43	1227	635	339	931	191	783	487	1079	117	1301	709	413	1005	265	857	561	1153	6	1
917	177	1361	769	473	1065	103	1287	695	399	991	251	843	547	1139	66	1250	658	362	954	214	806	510	1102	140	1324	732	436	1028	288	880	584	1176	29	1213	621	325
1343	751	455	1047	85	1269	677	381	973	233	825	529	1121	48	1232	640	344	936	196	788	492	1084	122	1306	714	418	1010	270	862	566	1158	11	1195	603	307	899	159
464	1056	94	1278	686	390	982	242	834	538	1130	57	1241	649	353	945	205	797	501	1093	131	1315	723	427	1019	279	871	575	1167	20	1204	612	316	908	168	1352	760
75	1259	667	371	963	223	815	519	1111	38	1222	630	334	926	186	778	482	1074	112	1296	704	408	1000	260	852	556	1148	1190	1185	593	297	889	149	1333	741	445	1037
702	406	998	258	850	554	1146	73	1257	665	369	961	221	813	517	1109	147	1331	739	443	1035	295	887	591	1183	36	1220	628	332	924	184	1368	776	480	1072	110	1294
980	240	832	536	1128	55	1239	647	351	943	203	795	499	1091	129	1313	721	425	1017	277	869	573	1165	18	1202	610	314	906	166	1350	758	462	1054	92	1276	684	388
841	545	1137	64	1248	656	360	952	212	804	508	1100	138	1322	730	434	1026	286	878	582	1174	27	1211	619	323	915	175	1359	767	471	1063	101	1285	693	397	989	249
1119	46	1230	638	342	934	194	786	490	1082	120	1304	712	416	1008	268	860	564	1156	9	1193	601	305	897	157	1341	749	453	1045	83	1267	675	379	971	231	823	527
1253	661	365	957	217	809	513	1105	143	1327	735	439	1031	291	883	587	1179	32	1216	624	328	920	180	1364	772	476	1068	106	1290	698	402	994	254	846	550	1142	69
347	939	199	791	495	1087	125	1309	717	421	1013	273	865	569	1161	14	1198	606	310	902	162	1346	754	458	1050	88	1272	680	384	976	236	828	532	1124	51	1235	643
208	800	504	1096	134	1318	726	430	1022	282	874	578	1170	23	1207	615	319	911	171	1355	763	467	1059	97	1281	689	393	985	245	837	541	1133	60	1244	652	356	948
485	1077	115	1299	707	411	1003	263	855	559	1151	4	1188	596	300	892	152	1336	744	448	1040	78	1262	670	374	966	226	818	522	1114	41	1225	633	337	929	189	781
145	1329	737	441	1033	293	885	589	1181	34	1218	626	330	922	182	1366	774	478	1070	108	1292	700	404	996	256	848	552	1144	71	1255	663	367	959	219	811	515	1107
719	423	1015	275	867	571	1163	16	1200	608	312	904	164	1348	756	460	1052	90	1274	682	386	978	238	830	534	1126	53	1237	645	349	941	201	793	497	1089	127	1311
1024	284	876	580	1172	25	1209	617	321	913	173	1357	765	469	1061	99	1283	691	395	987	247	839	543	1135	62	1246	654	358	950	210	802	506	1098	136	1320	728	432
858	562	1154	7	1191	599	303	895	155	1339	747	451	1043	81	1265	673	377	969	229	821	525	1117	44	1228	636	340	932	192	784	488	1080	118	1302	710	414	1006	266
1177	30	1214	622	326	918	178	1362	770	474	1066	104	1288	696	400	992	252	844	548	1140	67	1251	659	363	955	215	807	511	1103	141	1325	733	437	1029	289	881	585
1196	604	308	900	160	1344	752	456	1048	86	1270	678	382	974	234	826	530	1122	49	1233	641	345	937	197	789	493	1085	123	1307	715	419	1011	271	863	567	1159	12
317	909	169	1353	761	465	1057	95	1279	687	391	983	243	835	539	1131	58	1242	650	354	946	206	798	502	1094	132	1316	724	428	1020	280	872	576	1168	21	1205	613
150	1334	742	446	1038	76	1260	668	372	964	224	816	520	1112	39	1223	631	335	927	187	779	483	1075	113	1297	705	409	1001	261	853	557	1149	598	1186	594	298	890
775	479	1071	109	1293	701	405	997	257	849	553	1145	72	1256	664	368	960	220	812	516	1108	146	1330	738	442	1034	294	886	590	1182	35	1219	627	331	923	183	1367
1053	91	1275	683	387	979	239	831	535	1127	54	1238	646	350	942	202	794	498	1090	128	1312	720	424	1016	276	868	572	1164	17	1201	609	313	905	165	1349	757	461
1284	692	396	988	248	840	544	1136	63	1247	655	359	951	211	803	507	1099	137	1321	729	433	1025	285	877	581	1173	26	1210	618	322	914	174	1358	766	470	1062	100
378	970	230	822	526	1118	45	1229	637	341	933	193	785	489	1081	119	1303	711	415	1007	267	859	563	1155	8	1192	600	304	896	156	1340	748	452	1044	82	1266	674
253	845	549	1141	68	1252	660	364	956	216	808	512	1104	142	1326	734	438	1030	290	882	586	1178	31	1215	623	327	919	179	1363	771	475	1067	105	1289	697	401	993
531	1123	50	1234	642	346	938	198	790	494	1086	124	1308	716	420	1012	272	864	568	1160	13	1197	605	309	901	161	1345	753	457	1049	87	1271	679	383	975	235	827
59	1243	651	355	947	207	799	503	1095	133	1317	725	429	1021	281	873	577	1169	22	1206	614	318	910	170	1354	762	466	1058	96	1280	688	392	984	244	836	540	1132
632	336	928	188	780	484	1076	114	1298	706	410	1002	262	854	558	1150	1147	1187	595	299	891	151	1335	743	447	1039	77	1261	669	373	965	225	817	521	1113	40	1224
958	218	810	514	1106	144	1328	736	440	1032	292	884	588	1180	33	1217	625	329	921	181	1365	773	477	1069	107	1291	699	403	995	255	847	551	1143	70	1254	662	366
792	496	1088	126	1310	718	422	1014	274	866	570	1162	15	1199	607	311	903	163	1347	755	459	1051	89	1273	681	385	977	237	829	533	1125	52	1236	644	348	940	200
1097	135	1319	727	431	1023	283	875	579	1171	24	1208	616	320	912	172	1356	764	468	1060	98	1282	690	394	986	246	838	542	1134	61	1245	653	357	949	209	801	505
1300	708	412	1004	264	856	560	1152	5	1189	597	301	893	153	1337	745	449	1041	79	1263	671	375	967	227	819	523	1115	42	1226	634	338	930	190	782	486	1078	116
435	1027	287	879	583	1175	28	1212	620	324	916	176	1360	768	472	1064	102	1286	694	398	990	250	842	546	1138	65	1249	657	361	953	213	805	509	1101	139	1323	731
269	861	565	1157	10	1194	602	306	898	158	1342	750	454	1046	84	1268	676	380	972	232	824	528	1120	47	1231	639	343	935	195	787	491	1083	121	1305	713	417	1009
574	1166	19	1203	611	315	907	167	1351	759	463	1055	93	1277	685	389	981	241	833	537	1129	56	1240	648	352	944	204	796	500	1092	130	1314	722	426	1018	278	870
0	1184	592	296	888	148	1332	740	444	1036	74	1258	666	370	962	222	814	518	1110	37	1221	629	333	925	185	777	481	1073	111	1295	703	407	999	259	851	555	3

Table H.17 — 39 x 39 data

2	328	952	172	1420	796	484	1108	94	1342	718	406	1030	250	1498	874	562	1186	55	1303	679	367	991	211	1459	835	523	1147	133	1381	757	445	1069	289	913	601	1225	16	1
962	182	1430	806	494	1118	104	1352	728	416	1040	260	1508	884	572	1196	65	1313	689	377	1001	221	1469	845	533	1157	143	1391	767	455	1079	299	923	611	1235	26	1274	650	338
1410	786	474	1098	84	1332	708	396	1020	240	1488	864	552	1176	45	1293	669	357	981	201	1449	825	513	1137	123	1371	747	435	1059	279	903	591	1215	6	1254	630	318	942	162
499	1123	109	1357	733	421	1045	265	1513	889	577	1201	70	1318	694	382	1006	226	1474	850	538	1162	148	1396	772	460	1084	304	928	616	1240	31	1279	655	343	967	187	1435	811
89	1337	713	401	1025	245	1493	869	557	1181	50	1298	674	362	986	206	1454	830	518	1142	128	1376	752	440	1064	284	908	596	1220	11	1259	635	323	947	167	1415	791	479	1103
723	411	1035	255	1503	879	567	1191	60	1308	684	372	996	216	1464	840	528	1152	138	1386	762	450	1074	294	918	606	1230	21	1269	645	333	957	177	1425	801	489	1113	99	1347
1015	235	1483	859	547	1171	40	1288	664	352	976	196	1444	820	508	1132	118	1366	742	430	1054	274	898	586	1210	1264	1249	625	313	937	157	1405	781	469	1093	79	1327	703	391
1520	896	584	1208	77	1325	701	389	1013	233	1481	857	545	1169	155	1403	779	467	1091	311	935	623	1247	38	1286	662	350	974	194	1442	818	506	1130	116	1364	740	428	1052	272
565	1189	58	1306	682	370	994	214	1462	838	526	1150	136	1384	760	448	1072	292	916	604	1228	19	1267	643	331	955	175	1423	799	487	1111	97	1345	721	409	1033	253	1501	877
68	1316	692	380	1004	224	1472	848	536	1160	146	1394	770	458	1082	302	926	614	1238	29	1277	653	341	965	185	1433	809	497	1121	107	1355	731	419	1043	263	1511	887	575	1199
672	360	984	204	1452	828	516	1140	126	1374	750	438	1062	282	906	594	1218	9	1257	633	321	945	165	1413	789	477	1101	87	1335	711	399	1023	243	1491	867	555	1179	48	1296
1009	229	1477	853	541	1165	151	1399	775	463	1087	307	931	619	1243	34	1282	658	346	970	190	1438	814	502	1126	112	1360	736	424	1048	268	1516	892	580	1204	73	1321	697	385
1457	833	521	1145	131	1379	755	443	1067	287	911	599	1223	14	1262	638	326	950	170	1418	794	482	1106	92	1340	716	404	1028	248	1496	872	560	1184	53	1301	677	365	989	209
531	1155	141	1389	765	453	1077	297	921	609	1233	24	1272	648	336	960	180	1428	804	492	1116	102	1350	726	414	1038	258	1506	882	570	1194	63	1311	687	375	999	219	1467	843
121	1369	745	433	1057	277	901	589	1213	4	1252	628	316	940	160	1408	784	472	1096	82	1330	706	394	1018	238	1486	862	550	1174	43	1291	667	355	979	199	1447	823	511	1135
777	465	1089	309	933	621	1245	36	1284	660	348	972	192	1440	816	504	1128	114	1362	738	426	1050	270	1518	894	582	1206	75	1323	699	387	1011	231	1479	855	543	1167	153	1401
1070	290	914	602	1226	17	1265	641	329	953	173	1421	797	485	1109	95	1343	719	407	1031	251	1499	875	563	1187	56	1304	680	368	992	212	1460	836	524	1148	134	1382	758	446
924	612	1236	27	1275	651	339	963	183	1431	807	495	1119	105	1353	729	417	1041	261	1509	885	573	1197	66	1314	690	378	1002	222	1470	846	534	1158	144	1392	768	456	1080	300
1216	7	1255	631	319	943	163	1411	787	475	1099	85	1333	709	397	1021	241	1489	865	553	1177	46	1294	670	358	982	202	1450	826	514	1138	124	1372	748	436	1060	280	904	592
1280	656	344	968	188	1436	812	500	1124	110	1358	734	422	1046	266	1514	890	578	1202	71	1319	695	383	1007	227	1475	851	539	1163	149	1397	773	461	1085	305	929	617	1241	32
324	948	168	1416	792	480	1104	90	1338	714	402	1026	246	1494	870	558	1182	51	1299	675	363	987	207	1455	831	519	1143	129	1377	753	441	1065	285	909	597	1221	12	1260	636
178	1426	802	490	1114	100	1348	724	412	1036	256	1504	880	568	1192	61	1309	685	373	997	217	1465	841	529	1153	139	1387	763	451	1075	295	919	607	1231	22	1270	646	334	958
782	470	1094	80	1328	704	392	1016	236	1484	860	548	1172	41	1289	665	353	977	197	1445	821	509	1133	119	1367	743	431	1055	275	899	587	1211	640	1250	626	314	938	158	1406
1129	115	1363	739	427	1051	271	1519	895	583	1207	76	1324	700	388	1012	232	1480	856	544	1168	154	1402	778	466	1090	310	934	622	1246	37	1285	661	349	973	193	1441	817	505
1344	720	408	1032	252	1500	876	564	1188	57	1305	681	369	993	213	1461	837	525	1149	135	1383	759	447	1071	291	915	603	1227	18	1266	642	330	954	174	1422	798	486	1110	96
418	1042	262	1510	886	574	1198	67	1315	691	379	1003	223	1471	847	535	1159	145	1393	769	457	1081	301	925	613	1237	28	1276	652	340	964	184	1432	808	496	1120	106	1354	730
242	1490	866	554	1178	47	1295	671	359	983	203	1451	827	515	1139	125	1373	749	437	1061	281	905	593	1217	8	1256	632	320	944	164	1412	788	476	1100	86	1334	710	398	1022
891	579	1203	72	1320	696	384	1008	228	1476	852	540	1164	150	1398	774	462	1086	306	930	618	1242	33	1281	657	345	969	189	1437	813	501	1125	111	1359	735	423	1047	267	1515
1183	52	1300	676	364	988	208	1456	832	520	1144	130	1378	754	442	1066	286	910	598	1222	13	1261	637	325	949	169	1417	793	481	1105	91	1339	715	403	1027	247	1495	871	559
1310	686	374	998	218	1466	842	530	1154	140	1388	764	452	1076	296	920	608	1232	23	1271	647	335	959	179	1427	803	491	1115	101	1349	725	413	1037	257	1505	881	569	1193	62
354	978	198	1446	822	510	1134	120	1368	744	432	1056	276	900	588	1212	1209	1251	627	315	939	159	1407	783	471	1095	81	1329	705	393	1017	237	1485	861	549	1173	42	1290	666
230	1478	854	542	1166	152	1400	776	464	1088	308	932	620	1244	35	1283	659	347	971	191	1439	815	503	1127	113	1361	737	425	1049	269	1517	893	581	1205	74	1322	698	386	1010
834	522	1146	132	1380	756	444	1068	288	912	600	1224	15	1263	639	327	951	171	1419	795	483	1107	93	1341	717	405	1029	249	1497	873	561	1185	54	1302	678	366	990	210	1458
1156	142	1390	766	454	1078	298	922	610	1234	25	1273	649	337	961	181	1429	805	493	1117	103	1351	727	415	1039	259	1507	883	571	1195	64	1312	688	376	1000	220	1468	844	532
1370	746	434	1058	278	902	590	1214	5	1253	629	317	941	161	1409	785	473	1097	83	1331	707	395	1019	239	1487	863	551	1175	44	1292	668	356	980	200	1448	824	512	1136	122
459	1083	303	927	615	1239	30	1278	654	342	966	186	1434	810	498	1122	108	1356	732	420	1044	264	1512	888	576	1200	69	1317	693	381	1005	225	1473	849	537	1161	147	1395	771
283	907	595	1219	10	1258	634	322	946	166	1414	790	478	1102	88	1336	712	400	1024	244	1492	868	556	1180	49	1297	673	361	985	205	1453	829	517	1141	127	1375	751	439	1063
605	1229	20	1268	644	332	956	176	1424	800	488	1112	98	1346	722	410	1034	254	1502	878	566	1190	59	1307	683	371	995	215	1463	839	527	1151	137	1385	761	449	1073	293	917
0	1248	624	312	936	156	1404	780	468	1092	78	1326	702	390	1014	234	1482	858	546	1170	39	1287	663	351	975	195	1443	819	507	1131	117	1365	741	429	1053	273	897	585	3

Table H.18 — 41 x 41 data

2	332	1644	988	168	1480	824	496	1152	86	1398	742	414	1070	250	1562	906	578	1234	45	1357	701	373	1029	209	1521	865	537	1193	127	1439	783	455	1111	291	1603	947	619	1275	4	1
1677	1021	201	1513	857	529	1185	119	1431	775	447	1103	283	1595	939	611	1267	78	1390	734	406	1062	242	1554	898	570	1226	160	1472	816	488	1144	324	1636	980	652	1308	37	1349	693	365
181	1493	837	509	1165	99	1411	755	427	1083	263	1575	919	591	1247	58	1370	714	386	1042	222	1534	878	550	1206	140	1452	796	468	1124	304	1616	960	632	1288	17	1329	673	345	1657	1001
847	519	1175	109	1421	765	437	1093	273	1585	929	601	1257	68	1380	724	396	1052	232	1544	888	560	1216	150	1462	806	478	1134	314	1626	970	642	1298	27	1339	683	355	1667	1011	191	1503
1155	89	1401	745	417	1073	253	1565	909	581	1237	48	1360	704	376	1032	212	1524	868	540	1196	130	1442	786	458	1114	294	1606	950	622	1278	7	1319	663	335	1647	991	171	1483	827	499
1426	770	442	1098	278	1590	934	606	1262	73	1385	729	401	1057	237	1549	893	565	1221	155	1467	811	483	1139	319	1631	975	647	1303	32	1344	688	360	1672	1016	196	1508	852	524	1180	114
422	1078	258	1570	914	586	1242	53	1365	709	381	1037	217	1529	873	545	1201	135	1447	791	463	1119	299	1611	955	627	1283	12	1324	668	340	1652	996	176	1488	832	504	1160	94	1406	750
268	1580	924	596	1252	63	1375	719	391	1047	227	1539	883	555	1211	145	1457	801	473	1129	309	1621	965	637	1293	22	1334	678	350	1662	1006	186	1498	842	514	1170	104	1416	760	432	1088
903	575	1231	42	1354	698	370	1026	206	1518	862	534	1190	124	1436	780	452	1108	288	1600	944	616	1272	1316	1313	657	329	1641	985	165	1477	821	493	1149	83	1395	739	411	1067	247	1559
1270	81	1393	737	409	1065	245	1557	901	573	1229	163	1475	819	491	1147	327	1639	983	655	1311	40	1352	696	368	1680	1024	204	1516	860	532	1188	122	1434	778	450	1106	286	1598	942	614
1373	717	389	1045	225	1537	881	553	1209	143	1455	799	471	1127	307	1619	963	635	1291	20	1332	676	348	1660	1004	184	1496	840	512	1168	102	1414	758	430	1086	266	1578	922	594	1250	61
399	1055	235	1547	891	563	1219	153	1465	809	481	1137	317	1629	973	645	1301	30	1342	686	358	1670	1014	194	1506	850	522	1178	112	1424	768	440	1096	276	1588	932	604	1260	71	1383	727
215	1527	871	543	1199	133	1445	789	461	1117	297	1609	953	625	1281	10	1322	666	338	1650	994	174	1486	830	502	1158	92	1404	748	420	1076	256	1568	912	584	1240	51	1363	707	379	1035
896	568	1224	158	1470	814	486	1142	322	1634	978	650	1306	35	1347	691	363	1675	1019	199	1511	855	527	1183	117	1429	773	445	1101	281	1593	937	609	1265	76	1388	732	404	1060	240	1552
1204	138	1450	794	466	1122	302	1614	958	630	1286	15	1327	671	343	1655	999	179	1491	835	507	1163	97	1409	753	425	1081	261	1573	917	589	1245	56	1368	712	384	1040	220	1532	876	548
1460	804	476	1132	312	1624	968	640	1296	25	1337	681	353	1665	1009	189	1501	845	517	1173	107	1419	763	435	1091	271	1583	927	599	1255	66	1378	722	394	1050	230	1542	886	558	1214	148
456	1112	292	1604	948	620	1276	5	1317	661	333	1645	989	169	1481	825	497	1153	87	1399	743	415	1071	251	1563	907	579	1235	46	1358	702	374	1030	210	1522	866	538	1194	128	1440	784
325	1637	981	653	1309	38	1350	694	366	1678	1022	202	1514	858	530	1186	120	1432	776	448	1104	284	1596	940	612	1268	79	1391	735	407	1063	243	1555	899	571	1227	161	1473	817	489	1145
961	633	1289	18	1330	674	346	1658	1002	182	1494	838	510	1166	100	1412	756	428	1084	264	1576	920	592	1248	59	1371	715	387	1043	223	1535	879	551	1207	141	1453	797	469	1125	305	1617
1299	28	1340	684	356	1668	1012	192	1504	848	520	1176	110	1422	766	438	1094	274	1586	930	602	1258	69	1381	725	397	1053	233	1545	889	561	1217	151	1463	807	479	1135	315	1627	971	643
1320	664	336	1648	992	172	1484	828	500	1156	90	1402	746	418	1074	254	1566	910	582	1238	49	1361	705	377	1033	213	1525	869	541	1197	131	1443	787	459	1115	295	1607	951	623	1279	8
361	1673	1017	197	1509	853	525	1181	115	1427	771	443	1099	279	1591	935	607	1263	74	1386	730	402	1058	238	1550	894	566	1222	156	1468	812	484	1140	320	1632	976	648	1304	33	1345	689
997	177	1489	833	505	1161	95	1407	751	423	1079	259	1571	915	587	1243	54	1366	710	382	1038	218	1530	874	546	1202	136	1448	792	464	1120	300	1612	956	628	1284	13	1325	669	341	1653
1499	843	515	1171	105	1417	761	433	1089	269	1581	925	597	1253	64	1376	720	392	1048	228	1540	884	556	1212	146	1458	802	474	1130	310	1622	966	638	1294	23	1335	679	351	1663	1007	187
494	1150	84	1396	740	412	1068	248	1560	904	576	1232	43	1355	699	371	1027	207	1519	863	535	1191	125	1437	781	453	1109	289	1601	945	617	1273	660	1314	658	330	1642	986	166	1478	822
121	1433	777	449	1105	285	1597	941	613	1269	80	1392	736	408	1064	244	1556	900	572	1228	162	1474	818	490	1146	326	1638	982	654	1310	39	1351	695	367	1679	1023	203	1515	859	531	1187
757	429	1085	265	1577	921	593	1249	60	1372	716	388	1044	224	1536	880	552	1208	142	1454	798	470	1126	306	1618	962	634	1290	19	1331	675	347	1659	1003	183	1495	839	511	1167	101	1413
1095	275	1587	931	603	1259	70	1382	726	398	1054	234	1546	890	562	1218	152	1464	808	480	1136	316	1628	972	644	1300	29	1341	685	357	1669	1013	193	1505	849	521	1177	111	1423	767	439
1567	911	583	1239	50	1362	706	378	1034	214	1526	870	542	1198	132	1444	788	460	1116	296	1608	952	624	1280	9	1321	665	337	1649	993	173	1485	829	501	1157	91	1403	747	419	1075	255
608	1264	75	1387	731	403	1059	239	1551	895	567	1223	157	1469	813	485	1141	321	1633	977	649	1305	34	1346	690	362	1674	1018	198	1510	854	526	1182	116	1428	772	444	1100	280	1592	936
55	1367	711	383	1039	219	1531	875	547	1203	137	1449	793	465	1121	301	1613	957	629	1285	14	1326	670	342	1654	998	178	1490	834	506	1162	96	1408	752	424	1080	260	1572	916	588	1244
721	393	1049	229	1541	885	557	1213	147	1459	803	475	1131	311	1623	967	639	1295	24	1336	680	352	1664	1008	188	1500	844	516	1172	106	1418	762	434	1090	270	1582	926	598	1254	65	1377
1028	208	1520	864	536	1192	126	1438	782	454	1110	290	1602	946	618	1274	1271	1315	659	331	1643	987	167	1479	823	495	1151	85	1397	741	413	1069	249	1561	905	577	1233	44	1356	700	372
1553	897	569	1225	159	1471	815	487	1143	323	1635	979	651	1307	36	1348	692	364	1676	1020	200	1512	856	528	1184	118	1430	774	446	1102	282	1594	938	610	1266	77	1389	733	405	1061	241
549	1205	139	1451	795	467	1123	303	1615	959	631	1287	16	1328	672	344	1656	1000	180	1492	836	508	1164	98	1410	754	426	1082	262	1574	918	590	1246	57	1369	713	385	1041	221	1533	877
149	1461	805	477	1133	313	1625	969	641	1297	26	1338	682	354	1666	1010	190	1502	846	518	1174	108	1420	764	436	1092	272	1584	928	600	1256	67	1379	723	395	1051	231	1543	887	559	1215
785	457	1113	293	1605	949	621	1277	6	1318	662	334	1646	990	170	1482	826	498	1154	88	1400	744	416	1072	252	1564	908	580	1236	47	1359	703	375	1031	211	1523	867	539	1195	129	1441
1138	318	1630	974	646	1302	31	1343	687	359	1671	1015	195	1507	851	523	1179	113	1425	769	441	1097	277	1589	933	605	1261	72	1384	728	400	1056	236	1548	892	564	1220	154	1466	810	482
1610	954	626	1282	11	1323	667	339	1651	995	175	1487	831	503	1159	93	1405	749	421	1077	257	1569	913	585	1241	52	1364	708	380	1036	216	1528	872	544	1200	134	1446	790	462	1118	298
636	1292	21	1333	677	349	1661	1005	185	1497	841	513	1169	103	1415	759	431	1087	267	1579	923	595	1251	62	1374	718	390	1046	226	1538	882	554	1210	144	1456	800	472	1128	308	1620	964
0	1312	656	328	1640	984	164	1476	820	492	1148	82	1394	738	410	1066	246	1558	902	574	1230	41	1353	697	369	1025	205	1517	861	533	1189	123	1435	779	451	1107	287	1599	943	615	3

Table H.19 — 43 x 43 data

2	359	1735	1047	187	1563	875	531	1219	101	1477	789	445	1821	1133	273	1649	961	617	1305	58	1434	746	402	1778	1090	230	1606	918	574	1262	144	1520	832	488	1176	316	1692	1004	660	1348	15	1
1746	1058	198	1574	886	542	1230	112	1488	800	456	1832	1144	284	1660	972	628	1316	69	1445	757	413	1789	1101	241	1617	929	585	1273	155	1531	843	499	1187	327	1703	1015	671	1359	26	1402	714	370
176	1552	864	520	1208	90	1466	778	434	1810	1122	262	1638	950	606	1294	47	1423	735	391	1767	1079	219	1595	907	563	1251	133	1509	821	477	1165	305	1681	993	649	1337	4	1380	692	348	1724	1036
899	555	1243	125	1501	813	469	1845	1157	297	1673	985	641	1329	82	1458	770	426	1802	1114	254	1630	942	598	1286	168	1544	856	512	1200	340	1716	1028	684	1372	39	1415	727	383	1759	1071	211	1587
1222	104	1480	792	448	1824	1136	276	1652	964	620	1308	61	1437	749	405	1781	1093	233	1609	921	577	1265	147	1523	835	491	1179	319	1695	1007	663	1351	18	1394	706	362	1738	1050	190	1566	878	534
1491	803	459	1835	1147	287	1663	975	631	1319	72	1448	760	416	1792	1104	244	1620	932	588	1276	158	1534	846	502	1190	330	1706	1018	674	1362	29	1405	717	373	1749	1061	201	1577	889	545	1233	115
437	1813	1125	265	1641	953	609	1297	50	1426	738	394	1770	1082	222	1598	910	566	1254	136	1512	824	480	1168	308	1684	996	652	1340	7	1383	695	351	1727	1039	179	1555	867	523	1211	93	1469	781
1152	292	1668	980	636	1324	77	1453	765	421	1797	1109	249	1625	937	593	1281	163	1539	851	507	1195	335	1711	1023	679	1367	34	1410	722	378	1754	1066	206	1582	894	550	1238	120	1496	808	464	1840
1646	958	614	1302	55	1431	743	399	1775	1087	227	1603	915	571	1259	141	1517	829	485	1173	313	1689	1001	657	1345	12	1388	700	356	1732	1044	184	1560	872	528	1216	98	1474	786	442	1818	1130	270
625	1313	66	1442	754	410	1786	1098	238	1614	926	582	1270	152	1528	840	496	1184	324	1700	1012	668	1356	23	1399	711	367	1743	1055	195	1571	883	539	1227	109	1485	797	453	1829	1141	281	1657	969
44	1420	732	388	1764	1076	216	1592	904	560	1248	130	1506	818	474	1162	302	1678	990	646	1334	1391	1377	689	345	1721	1033	173	1549	861	517	1205	87	1463	775	431	1807	1119	259	1635	947	603	1291
773	429	1805	1117	257	1633	945	601	1289	171	1547	859	515	1203	343	1719	1031	687	1375	42	1418	730	386	1762	1074	214	1590	902	558	1246	128	1504	816	472	1848	1160	300	1676	988	644	1332	85	1461
1784	1096	236	1612	924	580	1268	150	1526	838	494	1182	322	1698	1010	666	1354	21	1397	709	365	1741	1053	193	1569	881	537	1225	107	1483	795	451	1827	1139	279	1655	967	623	1311	64	1440	752	408
247	1623	935	591	1279	161	1537	849	505	1193	333	1709	1021	677	1365	32	1408	720	376	1752	1064	204	1580	892	548	1236	118	1494	806	462	1838	1150	290	1666	978	634	1322	75	1451	763	419	1795	1107
913	569	1257	139	1515	827	483	1171	311	1687	999	655	1343	10	1386	698	354	1730	1042	182	1558	870	526	1214	96	1472	784	440	1816	1128	268	1644	956	612	1300	53	1429	741	397	1773	1085	225	1601
1284	166	1542	854	510	1198	338	1714	1026	682	1370	37	1413	725	381	1757	1069	209	1585	897	553	1241	123	1499	811	467	1843	1155	295	1671	983	639	1327	80	1456	768	424	1800	1112	252	1628	940	596
1521	833	489	1177	317	1693	1005	661	1349	16	1392	704	360	1736	1048	188	1564	876	532	1220	102	1478	790	446	1822	1134	274	1650	962	618	1306	59	1435	747	403	1779	1091	231	1607	919	575	1263	145
500	1188	328	1704	1016	672	1360	27	1403	715	371	1747	1059	199	1575	887	543	1231	113	1489	801	457	1833	1145	285	1661	973	629	1317	70	1446	758	414	1790	1102	242	1618	930	586	1274	156	1532	844
306	1682	994	650	1338	5	1381	693	349	1725	1037	177	1553	865	521	1209	91	1467	779	435	1811	1123	263	1639	951	607	1295	48	1424	736	392	1768	1080	220	1596	908	564	1252	134	1510	822	478	1166
1029	685	1373	40	1416	728	384	1760	1072	212	1588	900	556	1244	126	1502	814	470	1846	1158	298	1674	986	642	1330	83	1459	771	427	1803	1115	255	1631	943	599	1287	169	1545	857	513	1201	341	1717
1352	19	1395	707	363	1739	1051	191	1567	879	535	1223	105	1481	793	449	1825	1137	277	1653	965	621	1309	62	1438	750	406	1782	1094	234	1610	922	578	1266	148	1524	836	492	1180	320	1696	1008	664
1406	718	374	1750	1062	202	1578	890	546	1234	116	1492	804	460	1836	1148	288	1664	976	632	1320	73	1449	761	417	1793	1105	245	1621	933	589	1277	159	1535	847	503	1191	331	1707	1019	675	1363	30
352	1728	1040	180	1556	868	524	1212	94	1470	782	438	1814	1126	266	1642	954	610	1298	51	1427	739	395	1771	1083	223	1599	911	567	1255	137	1513	825	481	1169	309	1685	997	653	1341	8	1384	696
1067	207	1583	895	551	1239	121	1497	809	465	1841	1153	293	1669	981	637	1325	78	1454	766	422	1798	1110	250	1626	938	594	1282	164	1540	852	508	1196	336	1712	1024	680	1368	35	1411	723	379	1755
1561	873	529	1217	99	1475	787	443	1819	1131	271	1647	959	615	1303	56	1432	744	400	1776	1088	228	1604	916	572	1260	142	1518	830	486	1174	314	1690	1002	658	1346	13	1389	701	357	1733	1045	185
540	1228	110	1486	798	454	1830	1142	282	1658	970	626	1314	67	1443	755	411	1787	1099	239	1615	927	583	1271	153	1529	841	497	1185	325	1701	1013	669	1357	24	1400	712	368	1744	1056	196	1572	884
88	1464	776	432	1808	1120	260	1636	948	604	1292	45	1421	733	389	1765	1077	217	1593	905	561	1249	131	1507	819	475	1163	303	1679	991	647	1335	703	1378	690	346	1722	1034	174	1550	862	518	1206
815	471	1847	1159	299	1675	987	643	1331	84	1460	772	428	1804	1116	256	1632	944	600	1288	170	1546	858	514	1202	342	1718	1030	686	1374	41	1417	729	385	1761	1073	213	1589	901	557	1245	127	1503
1826	1138	278	1654	966	622	1310	63	1439	751	407	1783	1095	235	1611	923	579	1267	149	1525	837	493	1181	321	1697	1009	665	1353	20	1396	708	364	1740	1052	192	1568	880	536	1224	106	1482	794	450
289	1665	977	633	1321	74	1450	762	418	1794	1106	246	1622	934	590	1278	160	1536	848	504	1192	332	1708	1020	676	1364	31	1407	719	375	1751	1063	203	1579	891	547	1235	117	1493	805	461	1837	1149
955	611	1299	52	1428	740	396	1772	1084	224	1600	912	568	1256	138	1514	826	482	1170	310	1686	998	654	1342	9	1385	697	353	1729	1041	181	1557	869	525	1213	95	1471	783	439	1815	1127	267	1643
1326	79	1455	767	423	1799	1111	251	1627	939	595	1283	165	1541	853	509	1197	337	1713	1025	681	1369	36	1412	724	380	1756	1068	208	1584	896	552	1240	122	1498	810	466	1842	1154	294	1670	982	638
1433	745	401	1777	1089	229	1605	917	573	1261	143	1519	831	487	1175	315	1691	1003	659	1347	14	1390	702	358	1734	1046	186	1562	874	530	1218	100	1476	788	444	1820	1132	272	1648	960	616	1304	57
412	1788	1100	240	1616	928	584	1272	154	1530	842	498	1186	326	1702	1014	670	1358	25	1401	713	369	1745	1057	197	1573	885	541	1229	111	1487	799	455	1831	1143	283	1659	971	627	1315	68	1444	756
1078	218	1594	906	562	1250	132	1508	820	476	1164	304	1680	992	648	1336	1333	1379	691	347	1723	1035	175	1551	863	519	1207	89	1465	777	433	1809	1121	261	1637	949	605	1293	46	1422	734	390	1766
1629	941	597	1285	167	1543	855	511	1199	339	1715	1027	683	1371	38	1414	726	382	1758	1070	210	1586	898	554	1242	124	1500	812	468	1844	1156	296	1672	984	640	1328	81	1457	769	425	1801	1113	253
576	1264	146	1522	834	490	1178	318	1694	1006	662	1350	17	1393	705	361	1737	1049	189	1565	877	533	1221	103	1479	791	447	1823	1135	275	1651	963	619	1307	60	1436	748	404	1780	1092	232	1608	920
157	1533	845	501	1189	329	1705	1017	673	1361	28	1404	716	372	1748	1060	200	1576	888	544	1232	114	1490	802	458	1834	1146	286	1662	974	630	1318	71	1447	759	415	1791	1103	243	1619	931	587	1275
823	479	1167	307	1683	995	651	1339	6	1382	694	350	1726	1038	178	1554	866	522	1210	92	1468	780	436	1812	1124	264	1640	952	608	1296	49	1425	737	393	1769	1081	221	1597	909	565	1253	135	1511
1194	334	1710	1022	678	1366	33	1409	721	377	1753	1065	205	1581	893	549	1237	119	1495	807	463	1839	1151	291	1667	979	635	1323	76	1452	764	420	1796	1108	248	1624	936	592	1280	162	1538	850	506
1688	1000	656	1344	11	1387	699	355	1731	1043	183	1559	871	527	1215	97	1473	785	441	1817	1129	269	1645	957	613	1301	54	1430	742	398	1774	1086	226	1602	914	570	1258	140	1516	828	484	1172	312
667	1355	22	1398	710	366	1742	1054	194	1570	882	538	1226	108	1484	796	452	1828	1140	280	1656	968	624	1312	65	1441	753	409	1785	1097	237	1613	925	581	1269	151	1527	839	495	1183	323	1699	1011
0	1376	688	344	1720	1032	172	1548	860	516	1204	86	1462	774	430	1806	1118	258	1634	946	602	1290	43	1419	731	387	1763	1075	215	1591	903	559	1247	129	1505	817	473	1161	301	1677	989	645	3

Table H.20 — 45 x 45 data

2	370	1810	1090	190	1630	910	550	1990	1270	100	1540	820	460	1900	1180	280	1720	1000	640	1360	55	1495	775	415	1855	1135	235	1675	955	595	1315	145	1585	865	505	1945	1225	325	1765	1045	685	1405	10	1
1838	1118	218	1658	938	578	2018	1298	128	1568	848	488	1928	1208	308	1748	1028	668	1388	83	1523	803	443	1883	1163	263	1703	983	623	1343	173	1613	893	533	1973	1253	353	1793	1073	713	1433	38	1478	758	398
196	1636	916	556	1996	1276	106	1546	826	466	1906	1186	286	1726	1006	646	1366	61	1501	781	421	1861	1141	241	1681	961	601	1321	151	1591	871	511	1951	1231	331	1771	1051	691	1411	16	1456	736	376	1816	1096
927	567	2007	1287	117	1557	837	477	1917	1197	297	1737	1017	657	1377	72	1512	792	432	1872	1152	252	1692	972	612	1332	162	1602	882	522	1962	1242	342	1782	1062	702	1422	27	1467	747	387	1827	1107	207	1647
1984	1264	94	1534	814	454	1894	1174	274	1714	994	634	1354	49	1489	769	409	1849	1129	229	1669	949	589	1309	139	1579	859	499	1939	1219	319	1759	1039	679	1399	4	1444	724	364	1804	1084	184	1624	904	544
131	1571	851	491	1931	1211	311	1751	1031	671	1391	86	1526	806	446	1886	1166	266	1706	986	626	1346	176	1616	896	536	1976	1256	356	1796	1076	716	1436	41	1481	761	401	1841	1121	221	1661	941	581	2021	1301
829	469	1909	1189	289	1729	1009	649	1369	64	1504	784	424	1864	1144	244	1684	964	604	1324	154	1594	874	514	1954	1234	334	1774	1054	694	1414	19	1459	739	379	1819	1099	199	1639	919	559	1999	1279	109	1549
1920	1200	300	1740	1020	660	1380	75	1515	795	435	1875	1155	255	1695	975	615	1335	165	1605	885	525	1965	1245	345	1785	1065	705	1425	30	1470	750	390	1830	1110	210	1650	930	570	2010	1290	120	1560	840	480
277	1717	997	637	1357	52	1492	772	412	1852	1132	232	1672	952	592	1312	142	1582	862	502	1942	1222	322	1762	1042	682	1402	7	1447	727	367	1807	1087	187	1627	907	547	1987	1267	97	1537	817	457	1897	1177
1025	665	1385	80	1520	800	440	1880	1160	260	1700	980	620	1340	170	1610	890	530	1970	1250	350	1790	1070	710	1430	35	1475	755	395	1835	1115	215	1655	935	575	2015	1295	125	1565	845	485	1925	1205	305	1745
1363	58	1498	778	418	1858	1138	238	1678	958	598	1318	148	1588	868	508	1948	1228	328	1768	1048	688	1408	13	1453	733	373	1813	1093	193	1633	913	553	1993	1273	103	1543	823	463	1903	1183	283	1723	1003	643
1509	789	429	1869	1149	249	1689	969	609	1329	159	1599	879	519	1959	1239	339	1779	1059	699	1419	24	1464	744	384	1824	1104	204	1644	924	564	2004	1284	114	1554	834	474	1914	1194	294	1734	1014	654	1374	69
406	1846	1126	226	1666	946	586	1306	136	1576	856	496	1936	1216	316	1756	1036	676	1396	1450	1441	721	361	1801	1081	181	1621	901	541	1981	1261	91	1531	811	451	1891	1171	271	1711	991	631	1351	46	1486	766
1169	269	1709	989	629	1349	179	1619	899	539	1979	1259	359	1799	1079	719	1439	44	1484	764	404	1844	1124	224	1664	944	584	2024	1304	134	1574	854	494	1934	1214	314	1754	1034	674	1394	89	1529	809	449	1889
1687	967	607	1327	157	1597	877	517	1957	1237	337	1777	1057	697	1417	22	1462	742	382	1822	1102	202	1642	922	562	2002	1282	112	1552	832	472	1912	1192	292	1732	1012	652	1372	67	1507	787	427	1867	1147	247
618	1338	168	1608	888	528	1968	1248	348	1788	1068	708	1428	33	1473	753	393	1833	1113	213	1653	933	573	2013	1293	123	1563	843	483	1923	1203	303	1743	1023	663	1383	78	1518	798	438	1878	1158	258	1698	978
146	1586	866	506	1946	1226	326	1766	1046	686	1406	11	1451	731	371	1811	1091	191	1631	911	551	1991	1271	101	1541	821	461	1901	1181	281	1721	1001	641	1361	56	1496	776	416	1856	1136	236	1676	956	596	1316
894	534	1974	1254	354	1794	1074	714	1434	39	1479	759	399	1839	1119	219	1659	939	579	2019	1299	129	1569	849	489	1929	1209	309	1749	1029	669	1389	84	1524	804	444	1884	1164	264	1704	984	624	1344	174	1614
1952	1232	332	1772	1052	692	1412	17	1457	737	377	1817	1097	197	1637	917	557	1997	1277	107	1547	827	467	1907	1187	287	1727	1007	647	1367	62	1502	782	422	1862	1142	242	1682	962	602	1322	152	1592	872	512
343	1783	1063	703	1423	28	1468	748	388	1828	1108	208	1648	928	568	2008	1288	118	1558	838	478	1918	1198	298	1738	1018	658	1378	73	1513	793	433	1873	1153	253	1693	973	613	1333	163	1603	883	523	1963	1243
1040	680	1400	5	1445	725	365	1805	1085	185	1625	905	545	1985	1265	95	1535	815	455	1895	1175	275	1715	995	635	1355	50	1490	770	410	1850	1130	230	1670	950	590	1310	140	1580	860	500	1940	1220	320	1760
1437	42	1482	762	402	1842	1122	222	1662	942	582	2022	1302	132	1572	852	492	1932	1212	312	1752	1032	672	1392	87	1527	807	447	1887	1167	267	1707	987	627	1347	177	1617	897	537	1977	1257	357	1797	1077	717
1460	740	380	1820	1100	200	1640	920	560	2000	1280	110	1550	830	470	1910	1190	290	1730	1010	650	1370	65	1505	785	425	1865	1145	245	1685	965	605	1325	155	1595	875	515	1955	1235	335	1775	1055	695	1415	20
391	1831	1111	211	1651	931	571	2011	1291	121	1561	841	481	1921	1201	301	1741	1021	661	1381	76	1516	796	436	1876	1156	256	1696	976	616	1336	166	1606	886	526	1966	1246	346	1786	1066	706	1426	31	1471	751
1088	188	1628	908	548	1988	1268	98	1538	818	458	1898	1178	278	1718	998	638	1358	53	1493	773	413	1853	1133	233	1673	953	593	1313	143	1583	863	503	1943	1223	323	1763	1043	683	1403	8	1448	728	368	1808
1656	936	576	2016	1296	126	1566	846	486	1926	1206	306	1746	1026	666	1386	81	1521	801	441	1881	1161	261	1701	981	621	1341	171	1611	891	531	1971	1251	351	1791	1071	711	1431	36	1476	756	396	1836	1116	216
554	1994	1274	104	1544	824	464	1904	1184	284	1724	1004	644	1364	59	1499	779	419	1859	1139	239	1679	959	599	1319	149	1589	869	509	1949	1229	329	1769	1049	689	1409	14	1454	734	374	1814	1094	194	1634	914
1285	115	1555	835	475	1915	1195	295	1735	1015	655	1375	70	1510	790	430	1870	1150	250	1690	970	610	1330	160	1600	880	520	1960	1240	340	1780	1060	700	1420	25	1465	745	385	1825	1105	205	1645	925	565	2005
1532	812	452	1892	1172	272	1712	992	632	1352	47	1487	767	407	1847	1127	227	1667	947	587	1307	137	1577	857	497	1937	1217	317	1757	1037	677	1397	730	1442	722	362	1802	1082	182	1622	902	542	1982	1262	92
493	1933	1213	313	1753	1033	673	1393	88	1528	808	448	1888	1168	268	1708	988	628	1348	178	1618	898	538	1978	1258	358	1798	1078	718	1438	43	1483	763	403	1843	1123	223	1663	943	583	2023	1303	133	1573	853
1191	291	1731	1011	651	1371	66	1506	786	426	1866	1146	246	1686	966	606	1326	156	1596	876	516	1956	1236	336	1776	1056	696	1416	21	1461	741	381	1821	1101	201	1641	921	561	2001	1281	111	1551	831	471	1911
1742	1022	662	1382	77	1517	797	437	1877	1157	257	1697	977	617	1337	167	1607	887	527	1967	1247	347	1787	1067	707	1427	32	1472	752	392	1832	1112	212	1652	932	572	2012	1292	122	1562	842	482	1922	1202	302
639	1359	54	1494	774	414	1854	1134	234	1674	954	594	1314	144	1584	864	504	1944	1224	324	1764	1044	684	1404	9	1449	729	369	1809	1089	189	1629	909	549	1989	1269	99	1539	819	459	1899	1179	279	1719	999
82	1522	802	442	1882	1162	262	1702	982	622	1342	172	1612	892	532	1972	1252	352	1792	1072	712	1432	37	1477	757	397	1837	1117	217	1657	937	577	2017	1297	127	1567	847	487	1927	1207	307	1747	1027	667	1387
780	420	1860	1140	240	1680	960	600	1320	150	1590	870	510	1950	1230	330	1770	1050	690	1410	15	1455	735	375	1815	1095	195	1635	915	555	1995	1275	105	1545	825	465	1905	1185	285	1725	1005	645	1365	60	1500
1871	1151	251	1691	971	611	1331	161	1601	881	521	1961	1241	341	1781	1061	701	1421	26	1466	746	386	1826	1106	206	1646	926	566	2006	1286	116	1556	836	476	1916	1196	296	1736	1016	656	1376	71	1511	791	431
228	1668	948	588	1308	138	1578	858	498	1938	1218	318	1758	1038	678	1398	1395	1443	723	363	1803	1083	183	1623	903	543	1983	1263	93	1533	813	453	1893	1173	273	1713	993	633	1353	48	1488	768	408	1848	1128
985	625	1345	175	1615	895	535	1975	1255	355	1795	1075	715	1435	40	1480	760	400	1840	1120	220	1660	940	580	2020	1300	130	1570	850	490	1930	1210	310	1750	1030	670	1390	85	1525	805	445	1885	1165	265	1705
1323	153	1593	873	513	1953	1233	333	1773	1053	693	1413	18	1458	738	378	1818	1098	198	1638	918	558	1998	1278	108	1548	828	468	1908	1188	288	1728	1008	648	1368	63	1503	783	423	1863	1143	243	1683	963	603
1604	884	524	1964	1244	344	1784	1064	704	1424	29	1469	749	389	1829	1109	209	1649	929	569	2009	1289	119	1559	839	479	1919	1199	299	1739	1019	659	1379	74	1514	794	434	1874	1154	254	1694	974	614	1334	164
501	1941	1221	321	1761	1041	681	1401	6	1446	726	366	1806	1086	186	1626	906	546	1986	1266	96	1536	816	456	1896	1176	276	1716	996	636	1356	51	1491	771	411	1851	1131	231	1671	951	591	1311	141	1581	861
1249	349	1789	1069	709	1429	34	1474	754	394	1834	1114	214	1654	934	574	2014	1294	124	1564	844	484	1924	1204	304	1744	1024	664	1384	79	1519	799	439	1879	1159	259	1699	979	619	1339	169	1609	889	529	1969
1767	1047	687	1407	12	1452	732	372	1812	1092	192	1632	912	552	1992	1272	102	1542	822	462	1902	1182	282	1722	1002	642	1362	57	1497	777	417	1857	1137	237	1677	957	597	1317	147	1587	867	507	1947	1227	327
698	1418	23	1463	743	383	1823	1103	203	1643	923	563	2003	1283	113	1553	833	473	1913	1193	293	1733	1013	653	1373	68	1508	788	428	1868	1148	248	1688	968	608	1328	158	1598	878	518	1958	1238	338	1778	1058
0	1440	720	360	1800	1080	180	1620	900	540	1980	1260	90	1530	810	450	1890	1170	270	1710	990	630	1350	45	1485	765	405	1845	1125	225	1665	945	585	1305	135	1575	855	495	1935	1215	315	1755	1035	675	3

Table H.21 — 47 x 47 data

2	398	1902	1150	210	1714	962	586	2090	1338	116	1620	868	492	1996	1244	304	1808	1056	680	2184	1432	69	1573	821	445	1949	1197	257	1761	1009	633	2137	1385	163	1667	915	539	2043	1291	351	1855	1103	727	1479	22	1
1914	1162	222	1726	974	598	2102	1350	128	1632	880	504	2008	1256	316	1820	1068	692	2196	1444	81	1585	833	457	1961	1209	269	1773	1021	645	2149	1397	175	1679	927	551	2055	1303	363	1867	1115	739	1491	34	1538	786	410
198	1702	950	574	2078	1326	104	1608	856	480	1984	1232	292	1796	1044	668	2172	1420	57	1561	809	433	1937	1185	245	1749	997	621	2125	1373	151	1655	903	527	2031	1279	339	1843	1091	715	1467	10	1514	762	386	1890	1138
980	604	2108	1356	134	1638	886	510	2014	1262	322	1826	1074	698	2202	1450	87	1591	839	463	1967	1215	275	1779	1027	651	2155	1403	181	1685	933	557	2061	1309	369	1873	1121	745	1497	40	1544	792	416	1920	1168	228	1732
2084	1332	110	1614	862	486	1990	1238	298	1802	1050	674	2178	1426	63	1567	815	439	1943	1191	251	1755	1003	627	2131	1379	157	1661	909	533	2037	1285	345	1849	1097	721	1473	16	1520	768	392	1896	1144	204	1708	956	580
122	1626	874	498	2002	1250	310	1814	1062	686	2190	1438	75	1579	827	451	1955	1203	263	1767	1015	639	2143	1391	169	1673	921	545	2049	1297	357	1861	1109	733	1485	28	1532	780	404	1908	1156	216	1720	968	592	2096	1344
850	474	1978	1226	286	1790	1038	662	2166	1414	51	1555	803	427	1931	1179	239	1743	991	615	2119	1367	145	1649	897	521	2025	1273	333	1837	1085	709	1461	4	1508	756	380	1884	1132	192	1696	944	568	2072	1320	98	1602
2017	1265	325	1829	1077	701	2205	1453	90	1594	842	466	1970	1218	278	1782	1030	654	2158	1406	184	1688	936	560	2064	1312	372	1876	1124	748	1500	43	1547	795	419	1923	1171	231	1735	983	607	2111	1359	137	1641	889	513
301	1805	1053	677	2181	1429	66	1570	818	442	1946	1194	254	1758	1006	630	2134	1382	160	1664	912	536	2040	1288	348	1852	1100	724	1476	19	1523	771	395	1899	1147	207	1711	959	583	2087	1335	113	1617	865	489	1993	1241
1065	689	2193	1441	78	1582	830	454	1958	1206	266	1770	1018	642	2146	1394	172	1676	924	548	2052	1300	360	1864	1112	736	1488	31	1535	783	407	1911	1159	219	1723	971	595	2099	1347	125	1629	877	501	2005	1253	313	1817
2169	1417	54	1558	806	430	1934	1182	242	1746	994	618	2122	1370	148	1652	900	524	2028	1276	336	1840	1088	712	1464	7	1511	759	383	1887	1135	195	1699	947	571	2075	1323	101	1605	853	477	1981	1229	289	1793	1041	665
84	1588	836	460	1964	1212	272	1776	1024	648	2152	1400	178	1682	930	554	2058	1306	366	1870	1118	742	1494	37	1541	789	413	1917	1165	225	1729	977	601	2105	1353	131	1635	883	507	2011	1259	319	1823	1071	695	2199	1447
812	436	1940	1188	248	1752	1000	624	2128	1376	154	1658	906	530	2034	1282	342	1846	1094	718	1470	13	1517	765	389	1893	1141	201	1705	953	577	2081	1329	107	1611	859	483	1987	1235	295	1799	1047	671	2175	1423	60	1564
1952	1200	260	1764	1012	636	2140	1388	166	1670	918	542	2046	1294	354	1858	1106	730	1482	25	1529	777	401	1905	1153	213	1717	965	589	2093	1341	119	1623	871	495	1999	1247	307	1811	1059	683	2187	1435	72	1576	824	448
236	1740	988	612	2116	1364	142	1646	894	518	2022	1270	330	1834	1082	706	1458	1526	1505	753	377	1881	1129	189	1693	941	565	2069	1317	95	1599	847	471	1975	1223	283	1787	1035	659	2163	1411	48	1552	800	424	1928	1176
1033	657	2161	1409	187	1691	939	563	2067	1315	375	1879	1127	751	1503	46	1550	798	422	1926	1174	234	1738	986	610	2114	1362	140	1644	892	516	2020	1268	328	1832	1080	704	2208	1456	93	1597	845	469	1973	1221	281	1785
2138	1386	164	1668	916	540	2044	1292	352	1856	1104	728	1480	23	1527	775	399	1903	1151	211	1715	963	587	2091	1339	117	1621	869	493	1997	1245	305	1809	1057	681	2185	1433	70	1574	822	446	1950	1198	258	1762	1010	634
176	1680	928	552	2056	1304	364	1868	1116	740	1492	35	1539	787	411	1915	1163	223	1727	975	599	2103	1351	129	1633	881	505	2009	1257	317	1821	1069	693	2197	1445	82	1586	834	458	1962	1210	270	1774	1022	646	2150	1398
904	528	2032	1280	340	1844	1092	716	1468	11	1515	763	387	1891	1139	199	1703	951	575	2079	1327	105	1609	857	481	1985	1233	293	1797	1045	669	2173	1421	58	1562	810	434	1938	1186	246	1750	998	622	2126	1374	152	1656
2062	1310	370	1874	1122	746	1498	41	1545	793	417	1921	1169	229	1733	981	605	2109	1357	135	1639	887	511	2015	1263	323	1827	1075	699	2203	1451	88	1592	840	464	1968	1216	276	1780	1028	652	2156	1404	182	1686	934	558
346	1850	1098	722	1474	17	1521	769	393	1897	1145	205	1709	957	581	2085	1333	111	1615	863	487	1991	1239	299	1803	1051	675	2179	1427	64	1568	816	440	1944	1192	252	1756	1004	628	2132	1380	158	1662	910	534	2038	1286
1110	734	1486	29	1533	781	405	1909	1157	217	1721	969	593	2097	1345	123	1627	875	499	2003	1251	311	1815	1063	687	2191	1439	76	1580	828	452	1956	1204	264	1768	1016	640	2144	1392	170	1674	922	546	2050	1298	358	1862
1462	5	1509	757	381	1885	1133	193	1697	945	569	2073	1321	99	1603	851	475	1979	1227	287	1791	1039	663	2167	1415	52	1556	804	428	1932	1180	240	1744	992	616	2120	1368	146	1650	898	522	2026	1274	334	1838	1086	710
1548	796	420	1924	1172	232	1736	984	608	2112	1360	138	1642	890	514	2018	1266	326	1830	1078	702	2206	1454	91	1595	843	467	1971	1219	279	1783	1031	655	2159	1407	185	1689	937	561	2065	1313	373	1877	1125	749	1501	44
396	1900	1148	208	1712	960	584	2088	1336	114	1618	866	490	1994	1242	302	1806	1054	678	2182	1430	67	1571	819	443	1947	1195	255	1759	1007	631	2135	1383	161	1665	913	537	2041	1289	349	1853	1101	725	1477	20	1524	772
1160	220	1724	972	596	2100	1348	126	1630	878	502	2006	1254	314	1818	1066	690	2194	1442	79	1583	831	455	1959	1207	267	1771	1019	643	2147	1395	173	1677	925	549	2053	1301	361	1865	1113	737	1489	32	1536	784	408	1912
1700	948	572	2076	1324	102	1606	854	478	1982	1230	290	1794	1042	666	2170	1418	55	1559	807	431	1935	1183	243	1747	995	619	2123	1371	149	1653	901	525	2029	1277	337	1841	1089	713	1465	8	1512	760	384	1888	1136	196
602	2106	1354	132	1636	884	508	2012	1260	320	1824	1072	696	2200	1448	85	1589	837	461	1965	1213	273	1777	1025	649	2153	1401	179	1683	931	555	2059	1307	367	1871	1119	743	1495	38	1542	790	414	1918	1166	226	1730	978
1330	108	1612	860	484	1988	1236	296	1800	1048	672	2176	1424	61	1565	813	437	1941	1189	249	1753	1001	625	2129	1377	155	1659	907	531	2035	1283	343	1847	1095	719	1471	14	1518	766	390	1894	1142	202	1706	954	578	2082
1624	872	496	2000	1248	308	1812	1060	684	2188	1436	73	1577	825	449	1953	1201	261	1765	1013	637	2141	1389	167	1671	919	543	2047	1295	355	1859	1107	731	1483	26	1530	778	402	1906	1154	214	1718	966	590	2094	1342	120
472	1976	1224	284	1788	1036	660	2164	1412	49	1553	801	425	1929	1177	237	1741	989	613	2117	1365	143	1647	895	519	2023	1271	331	1835	1083	707	1459	774	1506	754	378	1882	1130	190	1694	942	566	2070	1318	96	1600	848
1267	327	1831	1079	703	2207	1455	92	1596	844	468	1972	1220	280	1784	1032	656	2160	1408	186	1690	938	562	2066	1314	374	1878	1126	750	1502	45	1549	797	421	1925	1173	233	1737	985	609	2113	1361	139	1643	891	515	2019
1807	1055	679	2183	1431	68	1572	820	444	1948	1196	256	1760	1008	632	2136	1384	162	1666	914	538	2042	1290	350	1854	1102	726	1478	21	1525	773	397	1901	1149	209	1713	961	585	2089	1337	115	1619	867	491	1995	1243	303
691	2195	1443	80	1584	832	456	1960	1208	268	1772	1020	644	2148	1396	174	1678	926	550	2054	1302	362	1866	1114	738	1490	33	1537	785	409	1913	1161	221	1725	973	597	2101	1349	127	1631	879	503	2007	1255	315	1819	1067
1419	56	1560	808	432	1936	1184	244	1748	996	620	2124	1372	150	1654	902	526	2030	1278	338	1842	1090	714	1466	9	1513	761	385	1889	1137	197	1701	949	573	2077	1325	103	1607	855	479	1983	1231	291	1795	1043	667	2171
1590	838	462	1966	1214	274	1778	1026	650	2154	1402	180	1684	932	556	2060	1308	368	1872	1120	744	1496	39	1543	791	415	1919	1167	227	1731	979	603	2107	1355	133	1637	885	509	2013	1261	321	1825	1073	697	2201	1449	86
438	1942	1190	250	1754	1002	626	2130	1378	156	1660	908	532	2036	1284	344	1848	1096	720	1472	15	1519	767	391	1895	1143	203	1707	955	579	2083	1331	109	1613	861	485	1989	1237	297	1801	1049	673	2177	1425	62	1566	814
1202	262	1766	1014	638	2142	1390	168	1672	920	544	2048	1296	356	1860	1108	732	1484	27	1531	779	403	1907	1155	215	1719	967	591	2095	1343	121	1625	873	497	2001	1249	309	1813	1061	685	2189	1437	74	1578	826	450	1954
1742	990	614	2118	1366	144	1648	896	520	2024	1272	332	1836	1084	708	1460	1457	1507	755	379	1883	1131	191	1695	943	567	2071	1319	97	1601	849	473	1977	1225	285	1789	1037	661	2165	1413	50	1554	802	426	1930	1178	238
653	2157	1405	183	1687	935	559	2063	1311	371	1875	1123	747	1499	42	1546	794	418	1922	1170	230	1734	982	606	2110	1358	136	1640	888	512	2016	1264	324	1828	1076	700	2204	1452	89	1593	841	465	1969	1217	277	1781	1029
1381	159	1663	911	535	2039	1287	347	1851	1099	723	1475	18	1522	770	394	1898	1146	206	1710	958	582	2086	1334	112	1616	864	488	1992	1240	300	1804	1052	676	2180	1428	65	1569	817	441	1945	1193	253	1757	1005	629	2133
1675	923	547	2051	1299	359	1863	1111	735	1487	30	1534	782	406	1910	1158	218	1722	970	594	2098	1346	124	1628	876	500	2004	1252	312	1816	1064	688	2192	1440	77	1581	829	453	1957	1205	265	1769	1017	641	2145	1393	171
523	2027	1275	335	1839	1087	711	1463	6	1510	758	382	1886	1134	194	1698	946	570	2074	1322	100	1604	852	476	1980	1228	288	1792	1040	664	2168	1416	53	1557	805	429	1933	1181	241	1745	993	617	2121	1369	147	1651	899
1305	365	1869	1117	741	1493	36	1540	788	412	1916	1164	224	1728	976	600	2104	1352	130	1634	882	506	2010	1258	318	1822	1070	694	2198	1446	83	1587	835	459	1963	1211	271	1775	1023	647	2151	1399	177	1681	929	553	2057
1845	1093	717	1469	12	1516	764	388	1892	1140	200	1704	952	576	2080	1328	106	1610	858	482	1986	1234	294	1798	1046	670	2174	1422	59	1563	811	435	1939	1187	247	1751	999	623	2127	1375	153	1657	905	529	2033	1281	341
729	1481	24	1528	776	400	1904	1152	212	1716	964	588	2092	1340	118	1622	870	494	1998	1246	306	1810	1058	682	2186	1434	71	1575	823	447	1951	1199	259	1763	1011	635	2139	1387	165	1669	917	541	2045	1293	353	1857	1105
0	1504	752	376	1880	1128	188	1692	940	564	2068	1316	94	1598	846	470	1974	1222	282	1786	1034	658	2162	1410	47	1551	799	423	1927	1175	235	1739	987	611	2115	1363	141	1645	893	517	2021	1269	329	1833	1081	705	3

Annex I
(normative)

ECC 000 - 140 character encodation schemes

This Annex provides details of the ASCII character set (ISO/IEC 646) used for one of the ECC 000 - 140 encodation schemes, and the four encodation schemes showing the mapping of the data character to the encodation scheme code value.

Table I.1 — Mapping of data character value to encodation scheme value

ASCII SET		ENCODATION SCHEME			
Character	**Decimal value**	**Base 11 code value**	**Base 27 code value**	**Base 37 code value**	**Base 41 code value**
NUL	0				
SOH	1				
STX	2				
ETX	3				
EOT	4				
ENQ	5				
ACK	6				
BEL	7				
BS	8				
HT	9				
LF	10				
VT	11				
FF	12				
CR	13				
SO	14				
SI	15				
DLE	16				
DC1	17				
DC2	18				
DC3	19				
DC4	20				
NAK	21				
SYN	22				
ETB	23				
CAN	24				
EM	25				

ASCII SET		ENCODATION SCHEME			
Character	**Decimal value**	**Base 11 code value**	**Base 27 code value**	**Base 37 code value**	**Base 41 code value**
SUB	26				
ESC	27				
FS	28				
GS	29				
RS	30				
US	31				
space	32	0	0	0	0
!	33				
“	34				
#	35				
$	36				
%	37				
&	38				
‘	39				
(	40				
)	41				
*	42				
+	43				
,	44				38
-	45				39
.	46				37
/	47				40
0	48	1		27	27
1	49	2		28	28
2	50	3		29	29
3	51	4		30	30
4	52	5		31	31
5	53	6		32	32
6	54	7		33	33
7	55	8		34	34
8	56	9		35	35
9	57	10		36	36
:	58				
;	59				
<	60				

ASCII SET		ENCODATION SCHEME			
Character	Decimal value	Base 11 code value	Base 27 code value	Base 37 code value	Base 41 code value
=	61				
>	62				
?	63				
@	64				
A	65		1	1	1
B	66		2	2	2
C	67		3	3	3
D	68		4	4	4
E	69		5	5	5
F	70		6	6	6
G	71		7	7	7
H	72		8	8	8
I	73		9	9	9
J	74		10	10	10
K	75		11	11	11
L	76		12	12	12
M	77		13	13	13
N	78		14	14	14
O	79		15	15	15
P	80		16	16	16
Q	81		17	17	17
R	82		18	18	18
S	83		19	19	19
T	84		20	20	20
U	85		21	21	21
V	86		22	22	22
W	87		23	23	23
X	88		24	24	24
Y	89		25	25	25
Z	90		26	26	26
[	91				
\	92				
]	93				
^	94				
-	95				

ASCII SET		ENCODATION SCHEME			
Character	Decimal value	Base 11 code value	Base 27 code value	Base 37 code value	Base 41 code value
‘	96				
a	97				
b	98				
c	99				
d	100				
e	101				
f	102				
g	103				
h	104				
i	105				
j	106				
k	107				
l	108				
m	109				
n	110				
o	111				
p	112				
q	113				
r	114				
s	115				
t	116				
u	117				
v	118				
w	119				
x	120				
y	121				
z	122				
{	123				
|	124				
}	125				
~	126				
DEL	127				

I.1 Base 11 encodation scheme

I.1.1 First stage procedure

The data characters shall be converted to their Base 11 code values using Table I.1 as the conversion table.

I.1.2 Second stage procedure

The following procedure shall be used to compact the Base 11 code values to a binary string.

a) Sub-divide the number of Base 11 characters into a sequence of six characters, from left to right. If less than six characters go to Step 5.

b) Assign the code values of the six Base 11 characters as C_1 to C_6 , where C_1 is the first character.

c) Carry out a Base 11 to Base 2 conversion to produce a sequence of 21 bits, using equation 6 of Table I.2.

d) Repeat from step a) as necessary.

e) When there are less than six characters, carry out a Base 11 to Base 2 conversion using the appropriate equation of Table I.2 which corresponds to the number of remaining Base 11 characters.

Table I.2 — Base 11 (Numeric) encodation equations

Number of data characters	Encodation equation	Bit length
1	C_1	4
2	$C_1 + C_2 * 11$	7
3	$C_1 + C_2 * 11 + C_3 * 11^2$	11
4	$C_1 + C_2 * 11 + C_3 * 11^2 + C_4 * 11^3$	14
5	$C_1 + C_2 * 11 + C_3 * 11^2 + C_4 * 11^3 + C_5 * 11^4$	18
6	$C_1 + C_2 * 11 + C_3 * 11^2 + C_4 * 11^3 + C_5 * 11^4 + C_6 * 11^5$	21

I.1.3 Example

Using the data character string: 123<space>45678 the complete Base 11 encodation process is shown in Figure I.1.

Data	1	2	3	<space>	4	5	6	7	8
Base 11 code value	2	3	4	0	5	6	7	8	9
Character position	C_1	C_2	C_3	C_4	C_5	C_6	C_1	C_2	C_3
Weight	1	11	121	1331	14641	161051	1	11	121
Product	2	33	484	0	73205	966306	7	88	1089
Decimal value	1040030						1184		
Binary string	011111101111010011110						10010100000		

Figure I.1 — Base 11 example

I.2 Base 27 encodation scheme

I.2.1 First stage procedure

The data characters shall be converted to their Base 27 code values using Table I.1 as the conversion table.

I.2.2 Second stage procedure

The following procedure shall be used to compact the Base 27 code values to a binary string.

a) Sub-divide the number of Base 27 characters into a sequence of five characters, from left to right. If less than five characters go to Step 5.

b) Assign the code values of the five Base 27 characters as C_1 to C_5, where C_1 is the first character.

c) Carry out a Base 27 to Base 2 conversion to produce a sequence of 24 bits, using equation 5 of Table I.3.

d) Repeat from step a) as necessary.

e) When there are less than five characters, carry out a Base 27 to Base 2 conversion using the appropriate equation of Table I.3 which corresponds to the number of remaining Base 27 characters.

Table I.3 — Base 27 (Upper-case Alphabetic) encodation equations

Number of data characters	Encodation equation	Bit length
1	C_1	5
2	$C_1 + C_2 * 27$	10
3	$C_1 + C_2 * 27 + C_3 * 27^2$	15
4	$C_1 + C_2 * 27 + C_3 * 27^2 + C_4 * 27^3$	20
5	$C_1 + C_2 * 27 + C_3 * 27^2 + C_4 * 27^3 + C_5 * 27^4$	24

I.2.3 Example

Using the data character string: DATA<space>MATRIX the complete Base 27 encodation process is shown in Figure I.2.

Data	**D**	**A**	**T**	**A**	**<space>**	**M**	**A**	**T**	**R**	**I**	**X**
Base 27 code value	4	1	20	1	0	13	1	20	18	9	24
Character position	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5	C_1
Weight	1	27	729	19683	531441	1	27	729	19683	531441	1
Product	4	27	14580	19683	0	13	27	14580	354294	4782969	24
Decimal Value	34294					5151883					24
Binary String	000000001000010111110110					010011101001110010001011					11000

Figure I.2 — Base 27 example

I.3 Base 37 encodation scheme

I.3.1 First stage procedure

The data characters shall be converted to their Base 37 code values using Table I.1 as the conversion table.

I.3.2 Second stage procedure

The following procedure shall be used to compact the Base 37 code values to a binary string.

a) Sub-divide the number of Base 37 characters into a sequence of four characters, from left to right. If less than four characters go to Step 5.

b) Assign the code values of the four Base 37 characters as C_1 to C_4, where C_1 is the first character.

c) Carry out a Base 37 to Base 2 conversion to produce a sequence of 21 bits, using equation 4 of Table I.4.

d) Repeat from step a) as necessary.

e) When there are less than four characters, carry out a Base 37 to Base 2 conversion using the equation (1 to 3) of Table I.4 which corresponds to the number of remaining Base 37 characters.

Table I.4 — Base 37 (Upper-case Alphanumeric) encodation equations

Number of data characters	Encodation equation	Bit length
1	C_1	6
2	$C_1 + C_2 * 37$	11
3	$C_1 + C_2 * 37 + C_3 * 37^2$	16
4	$C_1 + C_2 * 37 + C_3 * 37^2 + C_4 * 37^3$	21

I.3.3 Example

Using the data character string:

123ABCD89

the complete Base 37 encodation process is shown in Figure I.3.

Data	**1**	**2**	**3**	**A**	**B**	**C**	**D**	**8**	**9**
Base 37 code value	28	29	30	1	2	3	4	35	36
Character position	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1
Weight	1	37	1369	50653	1	37	1369	50653	1
Product	28	1073	41070	50653	2	111	5476	1772855	36
Decimal value	92824				1778444				36
Binary string	000010110101010011000				110110010001100001100				100100

Figure I.3 — Base 37 example

I.4 Base 41 encodation scheme

I.4.1 First stage procedure

The data characters shall be converted to their Base 41 code values using Table I.1 as the conversion table.

I.4.2 Second stage procedure

The following procedure shall be used to compact the Base 41 code values to a binary string.

a) Sub-divide the number of Base 41 characters into a sequence of four characters, from left to right. If less than four characters go to Step 5.

b) Assign the code values of the four Base 41 characters as C_1 to C_4, where C_1 is the first character.

c) Carry out a Base 41 to Base 2 conversion to produce a sequence of 22 bits, using equation 4 of Table I.5.

d) Repeat from step a) as necessary.

e) When there are less than four characters, carry out a Base 41 to Base 2 conversion using the appropriate equation of Table I.5 which corresponds to the number of remaining Base 41 characters.

Table I.5 — Base 41 (Upper-case alphanumeric + punctuation) encodation equations

Number of data characters	Encodation equation	Bit length
1	C_1	6
2	$C_1 + C_2 * 41$	11
3	$C_1 + C_2 * 41 + C_3 * 41^2$	17
4	$C_1 + C_2 * 41 + C_3 * 41^2 + C_4 * 41^3$	22

I.4.3 Example

Using the data character string:

AB/C123-X

the complete Base 41 encodation process is shown in Figure I.4.

Data	**A**	**B**	**/**	**C**	**1**	**2**	**3**	**-**	**X**
Base 41 code value	1	2	40	3	28	29	30	39	24
Character position	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1
Weight	1	41	1681	68921	1	41	1681	68921	1
Product	1	82	67240	206763	28	1189	50430	2687919	24
Decimal value	274086				2739566				24
Binary string	0001000010111010100110				1010011100110101101110				011000

Figure I.4 — Base 41 example

Annex J
(normative)

ECC 000 - 140 CRC algorithm

Following are two implementations for representing CRC.

J.1 CRC state machine

The CRC may be represented as a schematic, as illustrated in Figure J.1. After the data bits have been shifted through the state machine the resulting CRC is read out of the 16 memory registers (m) in the diagram (left most register is the MSB).

J.2 CRC polynomial

The CRC algorithm shall be the CCITT standard polynomial:

$$X^{16} + X^{12} + X^{5} + 1$$

With $X = 2$, the value of the polynomial shown as a 17 bit value is:

$$10001000000100001_{\text{base 2}}$$

The CRC is the remainder after dividing the data string by this value.

J.3 CRC 2-byte header

The CRC calculation headers, as defined in Table J.1, are used in the CRC operation as a prefix to the 8-bit byte values of the data characters. The CRC 2-byte header is shifted into the state machine prior to the calculation of the CRC.

Table J.1 — CRC calculation header

Format ID	Encodation scheme	CRC calculation header		
		MS Byte	LS Byte	Hex
1	Base 11	00000001	00000000	01 00
2	Base 27	00000010	00000000	02 00
3	Base 41	00000011	00000000	03 00
4	Base 37	00000100	00000000	04 00
5	ASCII	00000101	00000000	05 00
6	8-bit Byte	00000110	00000000	06 00

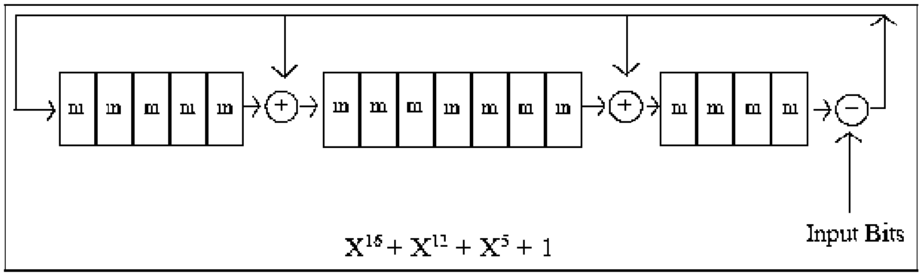


Figure J.1 — CRC algorithm schematic

Annex K
(normative)

ECC 000 - 140 error checking and correcting algorithms

K.1 ECC 000

This provides no error correction.

K.2 ECC 050

The error correction bit stream 'v' for ECC 050 shall be created by processing the unprotected bit stream 'u' through a state machine suitable for a convolutional code of the structure 4-3-3, as illustrated in Figure K.1.

K.3 ECC 080

The error correction bit stream 'v' for ECC 080 shall be created by processing the unprotected bit stream 'u' through a state machine suitable for a convolutional code of the structure 3-2-11, as illustrated in Figure K.2.

K.4 ECC 100

The error correction bit stream 'v' for ECC 100 shall be created by processing the unprotected bit stream 'u' through a state machine suitable for a convolutional code of the structure 2-1-15, as illustrated in Figure K.3.

K.5 ECC 140

The error correction bit stream 'v' for ECC 140 shall be created by processing the unprotected bit stream 'u' through a state machine suitable for a convolutional code of the structure 4-1-13, as illustrated in Figure K.4.

K.6 Processing the convolutional code

In the state machine circuit diagrams, the following notation is used:

$\boxed{m}$ represents a single bit storage register

$\boxed{+}$ represents a one bit binary adder which outputs the lowest bit. It is equivalent to an odd parity generator.

or such adjoining lines are connected

such intersecting lines are not connected

The state machine is operated as follows:

a) The memory storage registers (m) are filled with a zero value before starting the process.

b) An input cycle is performed, consisting of passing a user data bit through the input switch to a memory storage register (m) for each possible input switch position, i.e. for *k* bits.

c) Once a complete set of *k* input bits has been entered, an output cycle is performed. An output cycle consists of reading out an error corrected bit for each possible output switch position, i.e. for *n* bits. At each position, the output bit is computed by performing an XOR operation on the connected memory storage register values.

d) After one input and output cycle, a shift operation is performed by shifting all memory storage register values to the right by one position.

e) Steps b) through d) are repeated until all raw data bits have been input. At the end:

   1) Some zero bits may need to be added to the end of the last segment of input bits to ensure that *k* bits are input.

   2) Sufficient additional zero bits shall be input to ensure that the *m* memory storage registers shall all return to zero values. The output from steps e) 1) and e) 2) is part of the encoded data. The process is complete when all true data bits have passed through the last (rightmost) memory storage register.

K.7 Convolutional codes reference decode algorithm

The Fano algorithm can be used for error correction of data protected by convolutional codes. A basic description of the operation of the Fano algorithm is given in Lin and Costello (see Bibliography). The following guidelines should be used in constructing a convolutional coding decoder.

The start-up variable values must be as follows:

Backward Metric = maximum negative number

Current Metric = 0

Forward Metric = 0

Threshold = 0

The metric is computed by determining the number of bits that are different between the damaged block and the candidate match block:

Metric = (1 * correct bits) - (penalty * incorrect)

Table K1 presents values for the Single Bit Penalty and Delta which should be used when decoding each of the ECC levels.

Table K.1 — Fano algorithm coefficients

ECC level	Single bit penalty	Delta
050	31	20
080	16	11
100	8	6
140	4	1

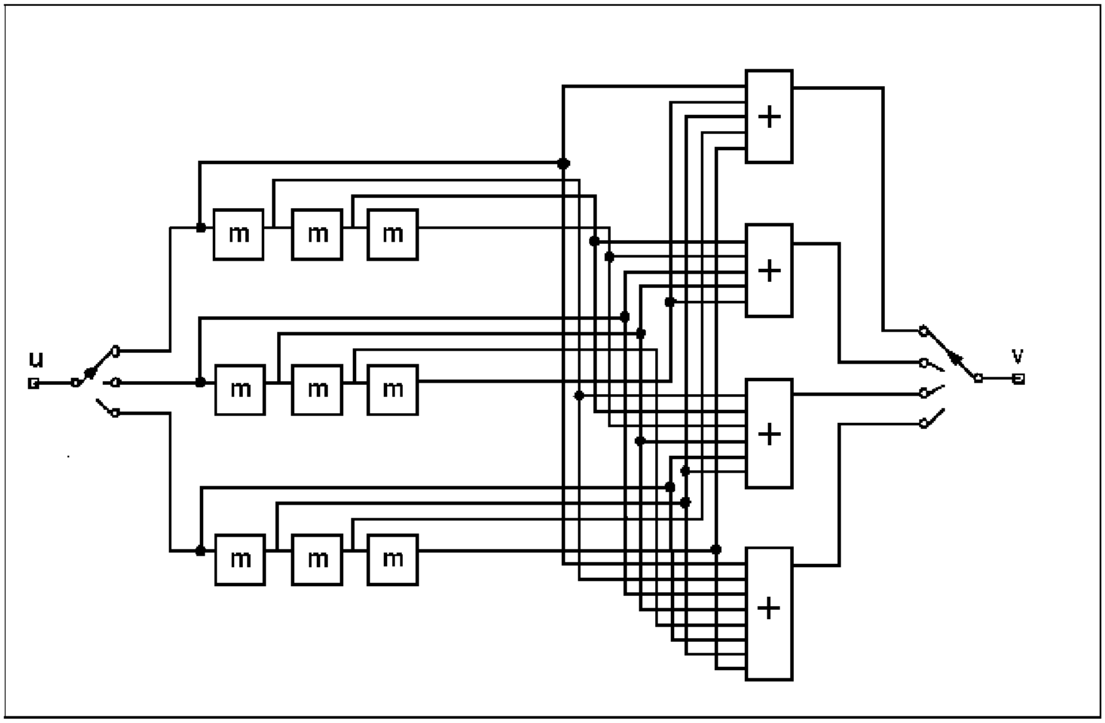


Figure K.1 — ECC 050; 4-3-3

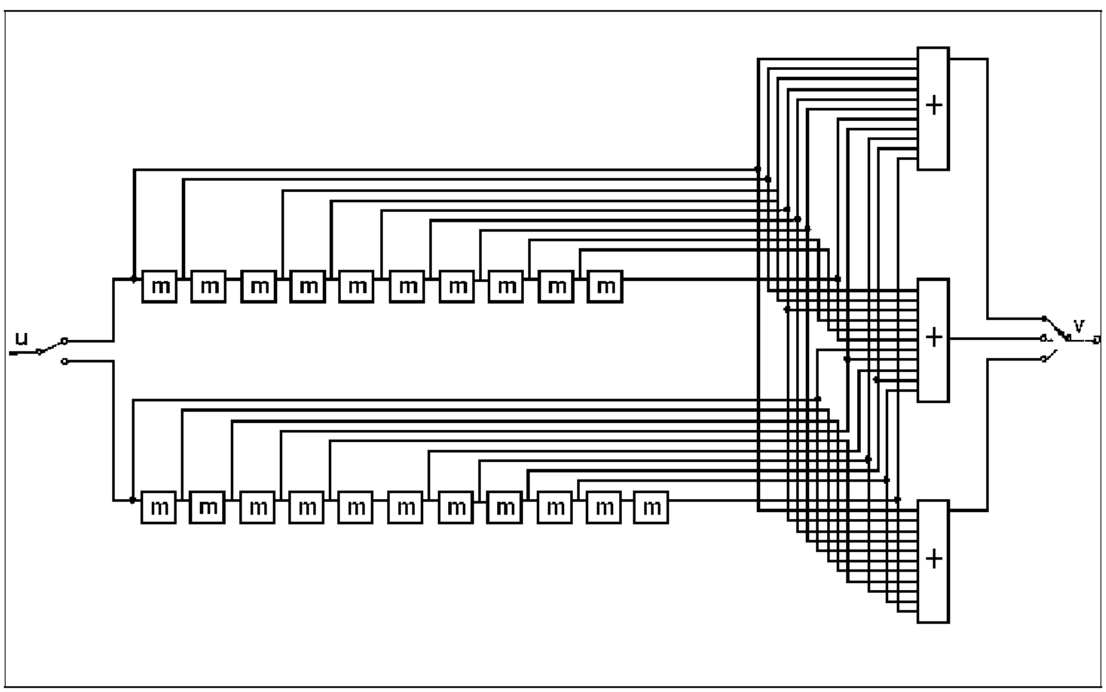


Figure K.2 — ECC 080; 3-2-11

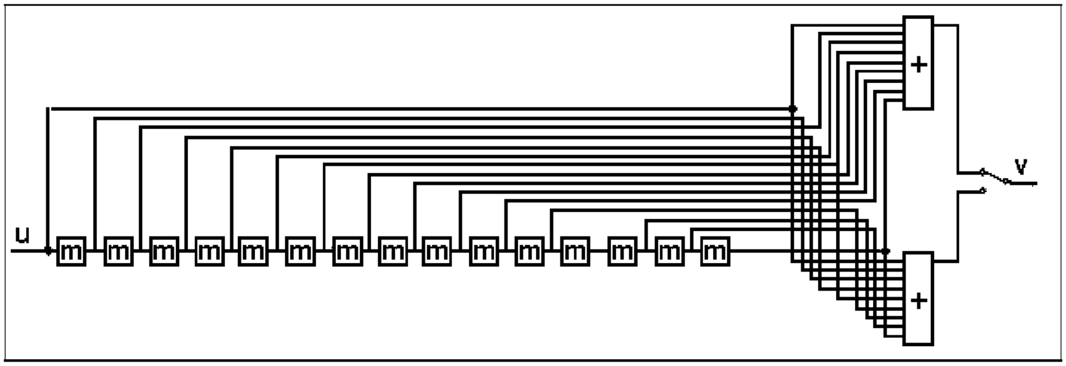


Figure K.3 — ECC 100; 2-1-15

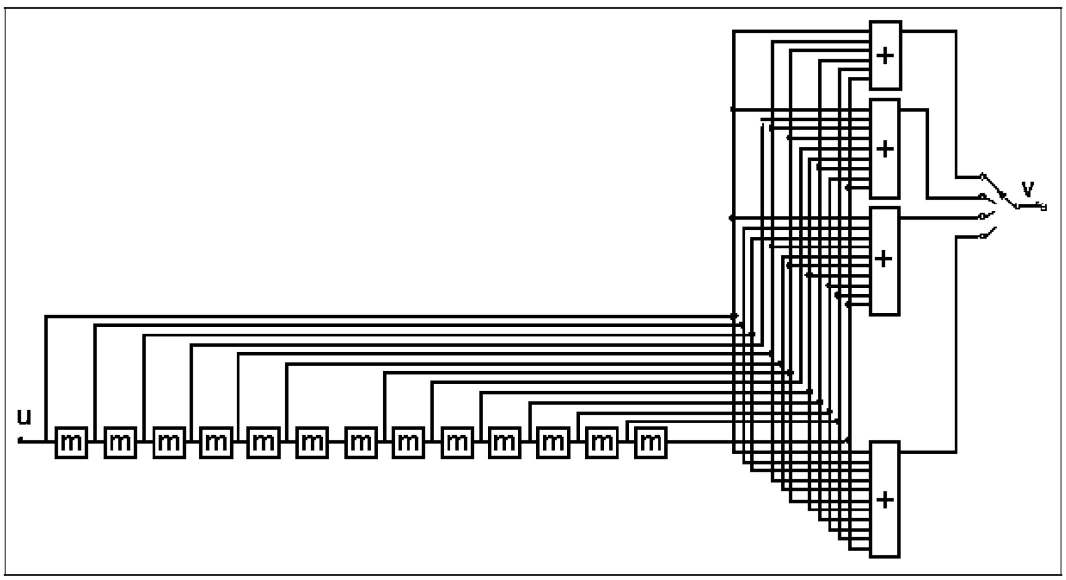


Figure K.4 — ECC 140; 4-1-13

Annex L
(normative)

ECC 000 - 140 Master Random Bit Stream (in hexadecimal)

```
(MSB)

05 ff c7 31 88 a8 83 9c 64 87 9f 64 b3 e0 4d 9c 80 29 3a 90

b3 8b 9e 90 45 bf f5 68 4b 08 cf 44 b8 d4 4c 5b a0 ab 72 52

1c e4 d2 74 a4 da 8a 08 fa a7 c7 dd 00 30 a9 e6 64 ab d5 8b

ed 9c 79 f8 08 d1 8b c6 22 64 0b 33 43 d0 80 d4 44 95 2e 6f

5e 13 8d 47 62 06 eb 80 82 c9 41 d5 73 8a 30 23 24 e3 7f b2

a8 0b ed 38 42 4c d7 b0 ce 98 bd e1 d5 e4 c3 1d 15 4a cf d1

1f 39 26 18 93 fc 19 b2 2d ab f2 6e a1 9f af d0 8a 2b a0 56

b0 41 6d 43 a4 63 f3 aa 7d af 35 57 c2 94 4a 65 0b 41 de b8

e2 30 12 27 9b 66 2b 34 5b b8 99 e8 28 71 d0 95 6b 07 4d 3c

7a b3 e5 29 b3 ba 8c cc 2d e0 c9 c0 22 ec 4c de f8 58 07 fc

19 f2 64 e2 c3 e2 d8 b9 fd 67 a0 bc f5 2e c9 49 75 62 82 27

10 f4 19 6f 49 f7 b3 84 14 ea eb e1 2a 31 ab 47 7d 08 29 ac

bb 72 fa fa 62 b8 c8 d3 86 89 95 fd df cc 9c ad f1 d4 6c 64

23 24 2a 56 1f 36 eb b7 d6 ff da 57 f4 50 79 08 0 (LSB)
```

Annex M
(normative)

Data Matrix print quality – symbology-specific aspects

Because of differences in symbology structures and reference decode algorithms, the effect of certain parameters on a symbol's reading performance may vary from one symbology to another. ISO/IEC 15415 provides for symbology specifications to define the grading of certain symbology-specific attributes. This annex therefore defines the method of grading Fixed Pattern Damage to be used in the application of ISO/IEC 15415 to Data Matrix.

M.1 Data Matrix Fixed Pattern Damage

M.1.1 Features to be assessed

The fixed pattern features to be assessed are contained in the one-module wide perimeter of the symbol and the quiet zone of a minimum of one module width (or more if specified by the application) surrounding the symbol. In larger symbols (square symbols 32 x 32 modules or larger, or rectangular symbols 8 x 32 or 12 x 36 or larger) with internal alignment patterns, the alignment pattern is also part of the fixed pattern. The left and lower side of the symbol should form a one-module wide solid "L" shape and the right and upper sides should consist of alternating dark and light single modules (known as the clock track). The alignment bars and internal clock track of the alignment pattern should similarly be a one-module wide solid bar or a series of alternating dark and light single modules respectively. The grading of Fixed Pattern Damage takes account not only of the total number of damaged modules but also of concentrations of damage.

M.1.2 Grading of the outside L of the fixed pattern

Damage to each side of the L shall be graded based on the modulation of the individual modules that compose it. These measurements are applied to the full length of the L sides and to the associated quiet zones.

Figure M.1 below indicates the four segments L1, L2, QZL1 and QZL2. Segment L1 is the vertical portion of the L and extends to the module in the quiet zone adjacent to the L corner. Segment L2 is the horizontal portion of the L and extends to the module in the quiet zone adjacent to the L corner. Segments QZL1 and QZL2 are the portions of the quiet zone adjacent to L1 and L2 respectively and extend one module beyond the end of L1 and L2 respectively, furthest from the corner and are shown shaded in Figure M.1. The corner module at the intersection of L1 and L2 is included in both segments, as is that at the intersection of QZL1 and QZL2.

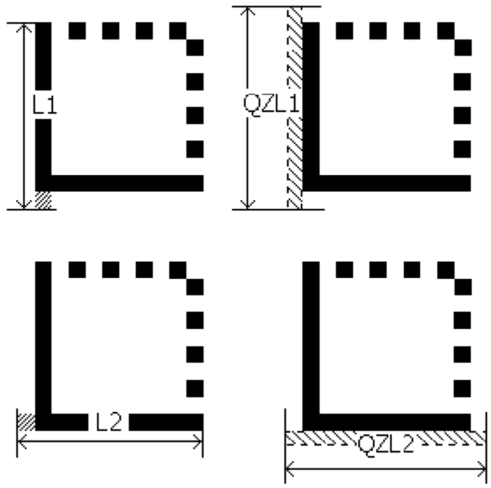


Figure M.1 — Outside L and corresponding quiet zone segments of fixed pattern

The procedure described below shall be applied to each segment in turn.

a) Find the modulation grade for each module based on the values in ISO/IEC 15415. Since the intended light or dark nature of the module is known, any module intended to be dark but the reflectance of which is above the global threshold, and any module intended to be light but the reflectance of which is below the global threshold shall be given modulation grade 0.

AC1 b) For each modulation grade level apply the parameter grade overlay technique described in ISO/IEC 15415:

1) For each side of the L (L1 and L2 in Figure M.1) and each quiet zone area (QZL1 and QZL2, adjacent to L1 and L2 respectively in Figure M.1), assume that all modules not achieving that grade or a higher grade are module errors, and derive a notional damage grade based on the grade thresholds shown in Table M.1. Take the lower of the modulation grade level and the notional damage grade.

2) The grade for each segment shall be the highest resulting grade for all modulation grade levels.

c) Additionally, for both square and rectangular symbols with more than one data region, repeat steps a) and b) above where L1 and L2 start with the module in the quiet zone and end at the module in the clock track area of the same data region and QZL1 and QZL2 consist of the quiet zone adjacent to these L1 and L2 segments as defined like Figure M.1. In other words treat the lower left data region as if it were a symbol with a single data region. If this grade is lower than that obtained from L1, L2, QZL1, and QZL2 in steps a) and b) then replace the grade obtained in steps a) and b) with this grade.

d) Additionally, for segments L1 and L2, verify that all gaps are separated by at least four correct modules and that no gaps are wider than three modules; if this test fails, the grade obtained from the above steps shall be reduced to 0 at that modulation grade level. AC1

AC1 e) AC1 The grade for Fixed Pattern Damage for the segment shall be the highest resulting grade for all modulation grade levels.

Table M.1 — Grade thresholds for notional damage

Percentage of modules damaged	Grade
0%	4
≤ 9%	3
≤ 13%	2
≤ 17%	1
> 17%	0

M.1.3 Grading of the clock track and adjacent solid area segments

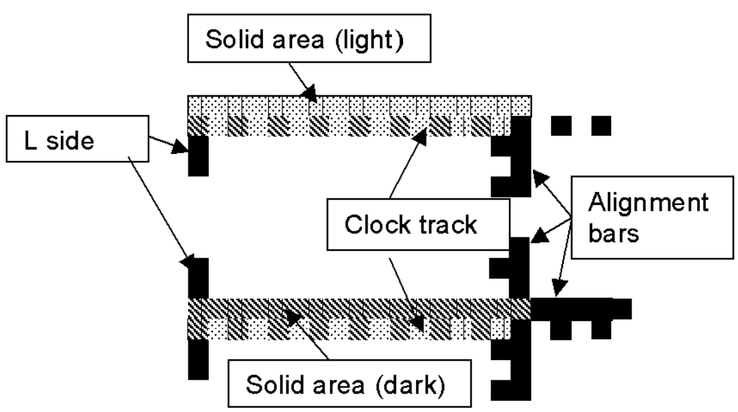
AC1 This section defines the measurement of damage to the internal alignment patterns (when present) and also external clock tracks and associated quiet zone areas. These tests are applied separately to each segment of the internal alignment patterns, the clock tracks, and associated quiet zone areas that bound the data region, or individual data regions of larger symbols. Each segment consists of a clock track portion and a solid area portion (which is part either of the quiet zone or of an internal alignment bar).

A clock track portion commences with a dark module in the L side or internal alignment bar perpendicular to it and continues to the light module preceding either the quiet zone or the next internal alignment bar.

A solid area portion with the alignment bar not adjacent to a quiet zone commences with the module adjacent to the first module of the associated clock track portion and continues to the module one past the last module of the associated clock track portion. Figure M.4 (a) illustrates the structures of these segments. The solid segments which correspond to portions of the external quiet zone are defined in this same way, as shown in Figure M.2.

A solid area portion with the alignment bar adjacent to a quiet zone commences with the module adjacent to the first module of the associated clock track portion and continues to the module adjacent to the last module of the associated clock track portion. Figure M.4 (b) illustrates the structures of these segments. AC1

NOTE In a symbol without internal alignment patterns, the external clock track segments extend for the full width or height of the symbol.



AC1 NOTE Figure M.2 depicts an internal alignment pattern segment which terminates at another internal alignment segment of the same color. AC1

Figure M.2 — Structure of external clock track segment and internal alignment pattern segment

a) For each external clock track segment or internal alignment pattern segment of a symbol (for multi-segment symbols), damage is measured according to the following procedure.

b) Transition ratio test.

On every clock track segment in the binarised image, both external (adjacent to the quiet zone) and internal (adjacent to the solid internal alignment bar), count the number of transitions in the clock track side, *Tc*, and the solid line side, *Ts*, and compute and grade the transition ratio *TR* as follows:

$Ts' = \text{Max}(0, Ts - 1)$

$TR = Ts' / Tc$

Table M.2 — Grading of Transition ratio

TR	Grade
$TR < 0{,}06$	4
$0{,}06 \leq TR < 0{,}08$	3
$0{,}08 \leq TR < 0{,}10$	2
$0{,}10 \leq TR < 0{,}12$	1
$TR \geq 0{,}12$	0

NOTE The end points between which transitions are counted are the intersections of grid lines plotted by the reference decode algorithm in the first and last modules of the clock track or solid area. See Figure M.3.

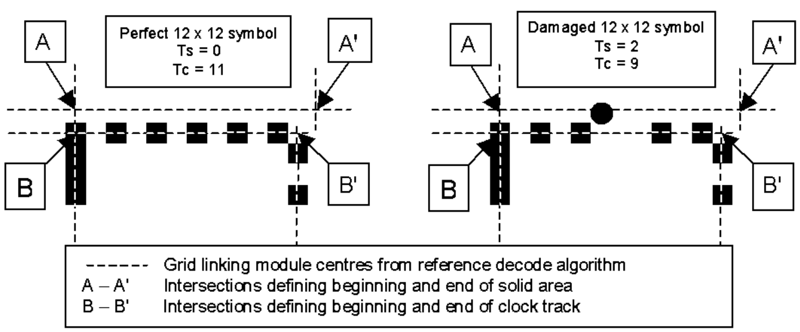


Figure M.3 — Transitions in perfect symbol (left) and damaged symbol (right)

c) Notional damage grade

Find the modulation grade for each module based on the values in ISO/IEC 15415. Since the intended light or dark nature of the module is known, any module intended to be dark but the reflectance of which is above the global threshold, and any module intended to be light but the reflectance of which is below the global threshold shall be given modulation grade 0.

d) For each modulation grade level:

Assume that all modules not achieving that grade or a higher grade are module errors, and derive a notional damage grade based on the following three assessments:

e) Clock track regularity test

For each segment of clock track, taking groups of five adjacent modules and progressing along the segment in steps of one module, verify that in any group of five adjacent modules no more than two are module errors; if this condition is met, the clock track regularity grade shall be 4, otherwise it shall be 0.

f) Clock track damage test

For each segment, count the number of incorrect modules in the clock track for the segment; the percentage *P* of incorrect modules over the length of the area shall result in the percentage damage grades shown in Table M.3.

g) Solid fixed pattern test

For each segment, count the number of incorrect modules in the solid area (internal alignment bar or external quiet zone area) adjacent to the clock track; the percentage *P* of incorrect modules over the length of the area shall result in the percentage damage grades shown in Table M.3.

Table M.3 — Grading of percentage damage to clock track segments and solid area segments

P	Grade
$P < 10\%$	4
$10\% \leq P < 15\%$	3
$15\% \leq P < 20\%$	2
$20\% \leq P < 25\%$	1
$P \geq 25\%$	0

h) At each grade level take the lowest of the modulation grade level, the clock track regularity grade, the clock track percentage damage grade, and the solid fixed pattern percentage damage grade.

i) The notional damage grade for the segment shall be the highest resulting grade for all modulation grade levels.

j) The Fixed Pattern Damage grade for the segment shall be the lower of the transition ratio grade and the notional damage grade.

k) The overall Fixed Pattern Damage grade for the clock track and adjacent solid area segments is the lowest of the grades obtained for each of the individual segments.

The shaded areas in Figure M.4 below show an example of an internal alignment pattern segment, which includes the clock track portion and solid area portion to which the transition ratio, regularity and solid fixed pattern tests are applied.

AC1

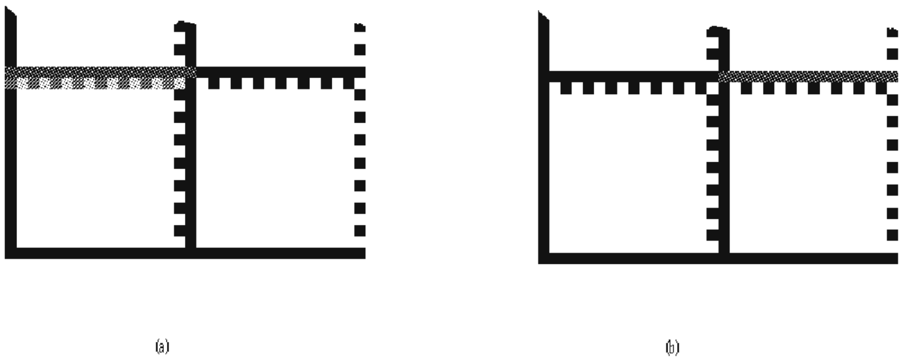


Figure M.4 — Internal alignment pattern segment which terminates at the external quiet zone AC1

The shaded areas in Figure M.5 below show an example of a segment of the external clock track and associated quiet zone to which the transition ratio, regularity and solid fixed pattern tests are applied.

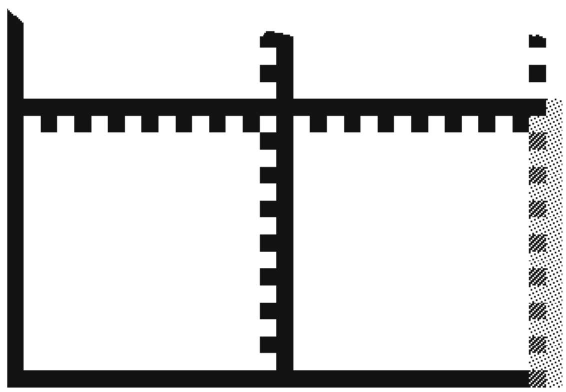


Figure M.5 — External clock track segment

AC1

Figure M.6 — Example showing the 37 modules graded for an L side of a 36×36 module symbol

EXAMPLE Figure M.6 shows an example based on grading the L1 segment of a 36 × 36 symbol, with *SC* = 89 % and *GT* = 51 %. The reflectance and modulation values, and modulation grade, are shown in Table M.4 for module 0 to 36 in the segment. The extended module on the quiet zone adjacent to the L corner is indicated as module 0.

Table M.4 — Example of modulation grading of 36-module segment

Module	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Reflectance (%)	84	15	13	13	13	9	11	84	11	10
MOD	74	80	86	86	86	94	90	(74)	90	92
MOD Grade	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**0**	**4**	**4**
Module		**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Reflectance (%)		9	11	70	13	12	15	11	11	11
MOD		94	90	(42)	86	88	80	90	90	90
MOD Grade		**4**	**4**	**0**	**4**	**4**	**4**	**4**	**4**	**4**
Module		**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
Reflectance (%)		27	11	14	10	12	50	12	11	14
MOD		54	90	83	92	88	2	88	90	83
MOD Grade		**4**	**4**	**4**	**4**	**4**	**0**	**4**	**4**	**4**
Module		**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**
Reflectance (%)		13	12	37	13	12	13	11	13	12
MOD		86	88	31	86	88	86	90	86	88
MOD Grade		**4**	**4**	**2**	**4**	**4**	**4**	**4**	**4**	**4**

NOTE Modules 0, 7 and 12 are clearly light; module 24, and to a lesser extent module 30, suffer from low modulation.

Based upon these values, the segment grading would be as shown in Table M.5.

Table M.5 — Example of grading segment

MOD grade level	No. of modules	Cum. No. Of modules	Remainder "damaged" modules	Damaged modules %	Notional damage grade	Lower of grade
4	33	33	4	10,8	2	2
3	0	33	4	10,8	2	2
2	1	34	3	8,1	3	2
1	0	34	3	8,1	3	1
0	3	37	0	0	4	0
Final Grade for segment – highest of last column						2

⟨AC1⟩

M.1.4 Calculation and grading of average grade

In addition to the assessment of the individual segments, a calculation of AG (average grade) is also made to take account of the cumulative effect of damage that is of relatively minor significance in individual segments but that affects several segments. This is based on averaging the grades for L1, L2, QZL1, QZL2 and the overall clock track and adjacent solid area segment grade

Once all segments have been graded, calculate the average grade AG:

AG = (Sum of the segment grades) / 5

Assign a grade to *AG* in accordance with Table M.6.

The Fixed Pattern Damage grade for the symbol shall be the lowest of the five segment grades and the grade for *AG*.

Table M.6 — Grading of *AG*

Mean of five segment grades	Grade
4	4
≥ 3,5	3
≥ 3,0	2
≥ 2,5	1
< 2,5	0

EXAMPLE 1

Assume that four of the five segments are graded 4, and one is graded 1. Then

(4 x 4) + (1 x 1) = 17

So *AG* = 17 / 5 = 3,4

From Table M.6, a mean of 3,4 will be graded 2. The lowest of the 6 grades is 1, and the symbol Fixed Pattern Damage grade, is therefore 1.

EXAMPLE 2

Assume that three of the five segments are graded 4, one is graded 3 and one is graded 1. Then

(3 x 4) + (1 x 3) + (1 x 1) = 16

So AG = 16 / 5 = 3,2

From Table M.6, a mean of 3,2 will be graded 2. The lowest of the 6 grades is1, and the symbol Fixed Pattern Damage grade is therefore1.

EXAMPLE 3

Assume that all of the five segments are graded 3. Then

5 x 3 = 15

So AG = 15 / 5 = 3,0

From Table M.6, a mean of 3,0 will be graded 2. The lowest of the 6 grades is 2, and the symbol Fixed Pattern Damage grade is therefore 2.

M.2 Scan grade

The scan grade shall be the lowest of the grades for the standard parameters evaluated according to ISO/IEC 15415 together with the grade for Fixed Pattern Damage evaluated in accordance with this Annex.

Annex N
(normative)

Symbology identifier

ISO/IEC 15424 provides a uniform methodology for reporting the symbology read, options set in the reader and any special features of the symbology encountered.

The symbology identifier for Data Matrix is:

]dm

where:

] is the symbology identifier flag (ASCII value 93)

d is the code character for the Data Matrix symbology

m is a modifier character with one of the values defined in Table N.1

Table N.1 — Symbology Identifier option values for Data Matrix

Option value	Option
0	ECC 000 - 140
1	ECC 200
2	ECC 200, FNC1 in 1st or 5th position
3	ECC 200, FNC1 in 2nd or 6th position
4	ECC 200 supporting ECI protocol
5	ECC 200, FNC1 in 1st or 5th position plus supporting ECI protocol
6	ECC 200, FNC1 in 2nd or 6th position plus supporting ECI protocol
NOTE (Permissible values of m: 0, 1, 2, 3, 4, 5, 6)	

Annex O
(informative)

ECC 200 encode example

In this example the user data to be encoded is “123456” (length of 6).

Step 1: Data encodation

The ASCII representation is:

data character:	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’
decimal:	49	50	51	52	53	54

ASCII encodation converts the above 6 characters to 3 bytes. This is done using the following formula for digit pairs.

Codeword = (numerical value of digit pairs) + 130

The details of this calculation are as follows.

“12” = 12 + 130 = 142

“34” = 34 + 130 = 164

“56” = 56 + 130 = 186

The data stream after data encodation is:

decimal: 142 164 186

Consulting Table 7, three data codewords fit exactly into a 10 x 10 symbol, and five error correction codewords need to be added. If the encoded data did not exactly fill a data region, then additional pads would have to be encoded.

Step 2: Error checking and correction

Error correction codewords are generated using the Reed-Solomon algorithm and appended to the encodation data stream. The resulting data stream is:

codeword:	1	2	3	4	5	6	7	8
decimal:	142	164	186	114	25	5	88	102
hex:	8E	A4	BA	72	19	05	58	66
	____data____/			_______check_______/				

Annex E describes the error correction process for ECC 200 and E.3 gives an example of a routine to perform the calculation of the error correction codewords.

Step 3: Module placement in matrix:

The final codewords from Step 2 are placed in the binary matrix as symbol characters according to the algorithm described in 5.8.1 (also see Figure F.1):

1	0	0	1	0	1	1	0
1	0	0	0	0	0	1	0
1	0	0	0	1	1	1	0
1	0	0	0	0	1	0	0
0	0	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
0	0	1	1	1	0	1	0

Figure O.1 — Module positioning in matrix

Step 4: Actual symbol

The final Data Matrix symbol is produced by adding the finder pattern modules and converting the binary ones to black and binary zeroes to white.

Figure O.2 — Final Data Matrix symbol encoding "123456"

Annex P
(informative)

Encoding data using the minimum symbol data characters for ECC 200

The same data may be represented by different Data Matrix symbols through the use of different code sets.

The following algorithm will usually produce the shortest codeword stream.

a) Start in ASCII encodation.

b) While in ASCII encodation:

   1) If the next data sequence is at least 2 consecutive digits, encode the next two digits as a double digit in ASCII mode.

   2) If the look-ahead test (starting at step j) indicates another mode, switch to that mode.

   3) If the Base 256 encodation mode has been indicated, encode the Latch to Base 256 encodation mode character followed by a currently undefined length byte; step G or step I will fill in the length field (this may require adding a second length byte).

   4) If the next data character is extended ASCII (greater than 127) encode it in ASCII mode first using the Upper Shift (value 235) character.

   5) Otherwise process the next data character in ASCII encodation.

c) While in C40 encodation:

   1) If the C40 encoding is at the point of starting a new double symbol character and if the look-ahead test (starting at step J) indicates another mode, switch to that mode.

   2) Otherwise process the next character in C40 encodation.

d) While in Text encodation:

   1) If the Text encoding is at the point of starting a new double symbol character and if the look-ahead test (starting at step J) indicates another mode, switch to that mode.

   2) Otherwise process the next character in Text encodation.

e) While in X12 encodation:

   1) If the X12 encoding is at the point of starting a new double symbol character and if the look-ahead test (starting at step J) indicates another mode, switch to that mode.

   2) Otherwise process the next character in X12 encodation.

f) While in EDIFACT (EDF) encodation:

   1) If the EDIFACT encoding is at the point of starting a new triple symbol character and if the look-ahead test (starting at step J) indicates another mode, switch to that mode.

   2) Otherwise process the next character in EDIFACT encodation.

g) While in Base 256 (B256) encodation:

   1) If the look-ahead test (starting at step J) indicates another mode, switch to that mode.

   2) Otherwise, process the next character in Base 256 encodation.

h) Repeat from step B until end of data.

i) At the end of data, if in Base 256 encodation, set the length to 0 (0 indicates that Base 256 encodation terminates the symbol).

The look-ahead test (Steps J through S):

The look-ahead test scans the data to be encoded to find the best mode.

j) Initialise the symbol character count for each mode:

   1) If the current mode is ASCII, initialise:

      ASCII count = 0,

      C40 count = 1,

      Text count = 1,

      X12 count = 1,

      EDF count = 1,

      B256 count = 1,25,

   otherwise initialise:

      ASCII count = 1,

      C40 count = 2,

      Text count = 2,

      X12 count = 2,

      EDF count = 2,

      B256 count = 2,25.

   2) If the current mode is C40 encodation, the C40 count = 0.

   3) If the current mode is Text encodation, the Text count = 0.

   4) If the current mode is X12 encodation, the X12 count = 0.

   5) If the current mode is EDIFACT encodation, the EDF count = 0.

   6) If the current mode is Base 256 encodation, the B256 count = 0.

k) If at the end of data:

   1) Round up all the counts to whole numbers.

   2) If the ASCII count is less than or equal to all the other counts, return from the test indicating ASCII encodation.

3) If the B256 count is less than all the other counts, return from the test indicating Base 256 encodation.

4) If the EDF count is less than all the other counts, return from the test indicating EDIFACT encodation.

5) If the Text count is less than all the other counts, return from the test indicating Text encodation.

6) If the X12 count is less than all the other counts, return from the test indicating X12 encodation.

7) Return from the test indicating C40 encodation.

l) Process the ASCII count:

1) If the data character is a digit, add 1/2 to the ASCII count.

2) If the data character is extended ASCII (greater than 127), round up and add 2 to the ASCII count.

3) Otherwise round up and add 1 to the ASCII count.

m) Process the C40 count:

1) If the data character is a native C40 character, add 2/3 to the C40 count.

2) If the data character is extended ASCII (greater than 127), add 8/3 to the C40 count.

3) Otherwise add 4/3 to the C40 count.

n) Process the Text count:

1) If the data character is a native Text character, add 2/3 to the Text count.

2) If the data character is extended ASCII (greater than 127), add 8/3 to the Text count.

3) Otherwise add 4/3 to the Text count.

o) Process the X12 count:

1) If the data character is a native X12 character, add 2/3 to the X12 count.

2) If the data character is extended ASCII (greater than 127), add 13/3 to the X12 count.

3) Otherwise add 10/3 to the X12 count.

p) Process the EDF count:

1) If the data character is a native EDF character, add 3/4 to the X12 count.

2) If the data character is extended ASCII (greater than 127), add 17/4 to the X12 count.

3) Otherwise add 13/4 to the X12 count.

q) Process the B256 count:

1) If the character is a Function character (FNC1, Structured Append, Reader Program, or Code Page), add 4 to the B256 count.

2) Otherwise add 1 to the B256 count.

r) If at least 4 data characters have been processed in this test loop:

1) If the ASCII count plus 1 is less than or equal to all the other counts, return from the test indicating ASCII encodation.

2) If the B256 count plus 1 is less than or equal to the ASCII count or less than the other counts, return from the test indicating Base 256 encodation.

3) If the EDF count plus 1 is less than all the other counts, return from the test indicating EDIFACT encodation.

4) If the Text count plus 1 is less than all the other counts, return from the test indicating Text encodation.

5) If the X12 count plus 1 is less than all the other counts, return from the test indicating X12 encodation.

6) If the C40 count plus 1 is less than the ASCII, B256, EDF, and Text counts:

i) If the C40 count is less than the X12 count, return from the test indicating C40 count.

ii) If the C40 count equals the X12 count:

I) If one of the three X12 terminator/ separator characters first occurs in the yet to be processed data before a non-X12 character, return from the test indicating X12 encodation.

II) Otherwise return with the C40 encodation.

s) Repeat from step k) until a return condition occurs.

Annex Q
(informative)

ECC 000 - 140 encode example using ECC 050

Q.1 Encode example

User data to be encoded: "AB12-X". This will be encoded in base 41 (format ID3)

Step 1: Data encodation

	sequence 1	sequence 2
a) break user data into 4-character sequences:		
	A B 1 2	- X
b) convert to Base 41 code values:		
	1 2 28 29	39 24
c) apply conversion equations:		
	2045860	1023
d) convert to binary bit stream:		
	0111110011011110100100	01111111111
e) reverse each sequence to create the final Encoded Data Bit Stream:		
	0010010111101100111110	11111111110

Step 2: Data prefix construction

a) The format ID field for base 41 is given from Table 11 (Section 5.4.1):

00010

b) The CRC field is computed as shown in Q.2, then it receives an MSB/LSB reversal to result in:

1001 1010 1010 1110

c) The length field must be 6 in binary with MSB/LSB reversal:

011000000

d) The final Unprotected Bit Stream is shown in Figure Q.1.

Step 3: Error checking and correction:

The Unprotected Bit Stream is broken into 3-bit input blocks in preparation for input to the ECC 050 state machine. Three extra input blocks of all zeros have been added to the input block list; this gives a total of 24 input blocks (see Figure Q.1). The number of extra zero blocks added is equal to the longest shift register path through the state machine for the ECC being used; for ECC 050, 3 zero blocks are added. The basic flow of all ECC state machines is as follows:

a) Zero the state machine registers

b) Switch in a new input block (MSB goes to position 1)

c) Compute the output values from all XOR gates

d) Switch out an output block (MSB comes from position 1)

Table Q.1 shows the values of all state machine elements during the process of performing convolutional coding on the 24 input blocks.

The final Protected Bit Stream (length = 96 bits) is:

0000 1010 1011 1111 1010 1010 1010 0000 0100 0011 0110 1000 0101 0001 1000 0000 1110 1010 1001 1010 1001 1000 0100 1010

Unprotected Bit Stream (Step 2)			
00010	1001101010101110	011000000	0010010111101100111110 11111111110
fmt3	CRC-16	len	encoded data
Unprotected Bit Stream broken into 3-bit blocks with three extra input blocks (Step 3)			
000 101 001 101 010 101 110 011 000 000 001 001 011 110 110 011 111 011 111 111 110 000 000 000			

Figure Q.1 — Unprotected Bit Stream from steps 2 and 3

Table Q.1 — Values of all registers during convolutional encoding

state machine cycle	Input 1 2 3	register 1A 1B 1C 2A 2B 2C 3A 3B 3C	output 1 2 3 4	state machine cycle	input 1 2 3	register 1A 1B 1C 2A 2B 2C 3A 3B 3C	output 1 2 3 4
1	0	000	0	13	0	000	0
	0	000	0		1	000	1
	0	000	0		1	110	0
			0				1
2	1	000	1	14	1	000	0
	0	000	0		1	100	0
	1	000	1		0	111	0
			0				1
3	0	100	1	15	1	100	1
	0	000	0		1	110	0
	1	100	1		0	011	0
			1				0
4	1	010	1	16	0	110	0
	0	000	1		1	111	0
	1	110	1		1	001	0
			1				0
5	0	101	1	17	1	011	1
	1	000	0		1	111	1
	0	111	1		1	100	1
			0				0
6	1	010	1	18	0	101	1
	0	100	0		1	111	0
	1	011	1		1	110	1
			0				0
7	1	101	1	19	1	010	1
	1	010	0		1	111	0
	0	101	1		1	111	0
			0				1
8	0	110	0	20	1	101	1
	1	101	0		1	111	0
	1	010	0		1	111	1
			0				0
9	0	011	0	21	1	110	1
	0	110	1		1	111	0
	0	101	0		0	111	0
			0				1
10	0	001	0	22	0	111	1
	0	011	0		0	111	0
	0	010	1		0	011	0
			1				0
11	0	000	0	23	0	011	0
	0	001	1		0	011	1
	1	001	1		0	001	0
			0				0
12	0	000	1	24	0	001	1
	0	000	0		0	001	0
	1	100	0		0	000	1
			0				0

Step 4: Header and trailer construction

The header contains the ECC bit field for 050 from Table 12 (6.6.1) with the bits reversed (MSB/LSB):

0111000000000111000 (length = 19 bits)

The trailer contains enough pad bits to make the Unrandomised Bit Stream fit exactly into a square matrix of the smallest size. There are 96 bits in the Protected Bit Stream and 19 bits in the header for a total of 115 bits.

A 13 x 13 data matrix has 11 x 11 information bits available (121 bits); this is the smallest matrix size able to contain 115 bits. There are 6 bits (121 - 115) that are set to zero. Therefore, the trailer is:

000000

The final Unrandomised Bit Stream is shown in Figure Q.2.

0111000000000111000 header	
000010101011111110101010101000000100001101101000010100011000000011101010100110101001100001001010 protected bit stream	000000 trailer

Figure Q.2 — Final Unrandomised Bit Stream

Step 5: Pattern randomising

Partition the Unrandomised Bit Stream into 4-bit nibbles for easy XORing:

0111 0000 0000 0111 0000 0001 0101 0111 1111 0101 0101 0100 0000 1000 0110 1101 0000 1010 0011 0000 0001 1101 0101 0011 0101 0011 0000 1001 0100 0000 0

Get the required number (121) of random bits from the Master Random Bit Stream (Annex L):

(05, FF, C7, 31, 88, A8, 83, 9C, 64, 87, 9F, 64, B3, E0, 4D, first bit of 9C) =

0000 0101 1111 1111 1100 0111 0011 0001 1000 1000 1010 1000 1000 0011 1001 1100 0110 0100 1000 0111 1001 1111 0110 0100 1011 0011 1110 0000 0100 1101 1

Produce the Randomised Bit Stream by XORing the input with the random bits:

0111 0101 1111 1000 1100 0110 0110 0110 0111 1101 1111 1100 1000 1011 1111 0001 0110 1110 1011 0111 1000 0010 0011 0111 1110 0000 1110 1001 0000 1101 1

Step 6: Module placement in matrix

Using the Data Module Placement Grid for this matrix size, the data modules are placed into the binary matrix data area:

1 1 0 1 0 0 1 1 0 0 1

1 0 0 1 0 1 0 1 1 0 1

1 0 1 1 1 0 0 1 0 1 0

1 1 0 1 1 1 0 1 0 1 0

0 1 1 0 0 0 0 1 1 0 0

1 1 1 0 1 0 0 1 1 0 1

0 0 1 0 0 1 1 1 1 1 0

1 0 1 0 1 1 1 1 0 0 1

0 1 1 1 1 1 0 1 0 1 0

1 0 0 1 0 0 1 1 1 1 0

0 0 1 1 0 1 1 0 1 1 1

After adding the finder pattern modules, the final binary matrix is produced:

1 0 1 0 1 0 1 0 1 0 1 0 1

1 1 1 0 1 0 0 1 1 0 0 1 0

1 1 0 0 1 0 1 0 1 1 0 1 1

1 1 0 1 1 1 0 0 1 0 1 0 0

1 1 1 0 1 1 1 0 1 0 1 0 1

1 0 1 1 0 0 0 0 1 1 0 0 0

1 1 1 1 0 1 0 0 1 1 0 1 1

1 0 0 1 0 0 1 1 1 1 1 0 0

1 1 0 1 0 1 1 1 1 0 0 1 1

1 0 1 1 1 1 1 0 1 0 1 0 0

1 1 0 0 1 0 0 1 1 1 1 0 1

1 0 0 1 1 0 1 1 0 1 1 1 0

1 1 1 1 1 1 1 1 1 1 1 1 1

Q.2 CRC calculation for example

Construct the bit stream for input to the CRC algorithm. This consists of the CRC 2-byte header followed by the original user data. The CRC 2-byte header from Annex J, Table J.1 for format 3 is:

0000 0011 0000 0000.

The original user data is:

A B 1 2 - X

0100 0001, 0100 0010, 0011 0001, 0011 0010, 0010 1101, 0101 1000

The complete pre-CRC bit stream before byte reversal is:

0000 0011, 0000 0000, 0100 0001, 0100 0010, 0011 0001, 0011 0010, 0010 1101, 0101 1000

The complete pre-CRC bit stream after byte reversal is: (64 bits)

1100 0000, 0000 0000, 1000 0010, 0100 0010, 1000 1100, 0100 1100, 1011 0100, 0001 1010

This bit stream is input to the CRC state machine shown in Table Q.2. The CRC MSB is in the left-most shift register, so the final computed CRC value is 01110 1010101 1001 when read directly from the state machine. Parsing into 4-bit nibbles yields: 0111, 0101, 0101, 1001 which is the CRC field value used in Annex Q, Step 2b.

Table Q.2 — Value of all registers during CRC calculation

Machine cycle	XOR register bits 1-5	gate out3	XOR register bits 6-12	gate out2	register bits 13-16	input bit	XOR gate out1
start-up	00000	1	0000000	1	0000	1	1
1	10000	1	1000000	1	1000	1	1
2	11000	0	1100000	0	1100	0	0
3	01100	0	0110000	0	0110	0	0
4	00110	1	0011000	1	0011	0	1
5	10011	0	1001100	1	1001	0	1
6	11001	1	0100110	0	1100	0	0
7	01100	0	1010011	1	0110	0	0
8	00110	1	0101001	0	1011	0	1
9	10011	0	1010100	1	0101	0	1
10	11001	1	0101010	0	1010	0	0
11	01100	1	1010101	0	0101	0	1
12	10110	0	1101010	0	0010	0	0
13	01011	0	0110101	0	0001	0	1
14	10101	1	0011010	0	0000	0	0
15	01010	0	1001101	1	0000	0	0
16	00101	0	0100110	1	1000	1	1
17	10010	0	0010011	1	1100	0	0

Machine cycle	XOR register bits 1-5	gate out3	XOR register bits 6-12	gate out2	register bits 13-16	input bit	XOR gate out1
18	01001	1	0001001	1	1110	0	0
19	00100	1	1000100	1	1111	0	1
20	10010	1	1100010	1	1111	0	1
21	11001	0	1110001	0	1111	0	1
22	11100	0	0111000	0	0111	1	0
23	01110	1	0011100	1	0011	0	1
24	10111	0	1001110	1	1001	0	1
25	11011	0	0100111	0	1100	1	1
26	11101	1	0010011	1	0110	0	0
27	01110	1	1001001	0	1011	0	1
28	10111	0	1100100	1	0101	0	1
29	11011	1	0110010	0	1010	0	0
30	01101	1	1011001	1	0101	1	0
31	00110	0	1101100	0	1010	0	0
32	00011	1	0110110	0	0101	1	0
33	00001	1	1011011	1	0010	0	0
34	00000	1	1101101	0	1001	0	1
35	10000	0	1110110	0	0100	0	0
36	01000	1	0111011	0	0010	1	1
37	10100	0	1011101	1	0001	1	0
38	01010	0	0101110	0	1000	0	0
39	00101	1	0010111	1	0100	0	0
40	00010	0	1001011	1	1010	0	0
41	00001	1	0100101	1	1101	1	0
42	00000	0	1010010	0	1110	0	0
43	00000	1	0101001	0	0111	0	1
44	10000	0	1010100	0	0011	1	0
45	01000	0	0101010	0	0001	1	0
46	00100	0	0010101	1	0000	0	0
47	00010	0	0001010	0	1000	0	0
48	00001	0	0000101	0	0100	1	1
49	10000	0	0000010	0	0010	0	0
50	01000	0	0000001	1	0001	1	0
51	00100	1	0000000	1	1000	1	1
52	10010	0	1000000	0	1100	0	0
53	01001	0	0100000	1	0110	1	1

Machine cycle	XOR register bits 1-5	gate out3	XOR register bits 6-12	gate out2	register bits 13-16	input bit	XOR gate out1
54	10100	1	0010000	1	1011	0	1
55	11010	1	1001000	1	1101	0	1
56	11101	1	1100100	0	1110	0	0
57	01110	1	1110010	1	0111	0	1
58	10111	0	1111001	0	1011	0	1
59	11011	1	0111100	0	0101	1	0
60	01101	0	1011110	1	0010	1	1
61	10110	1	0101111	0	1001	0	1
62	11011	0	1010111	0	0100	1	1
63	11101	1	0101011	1	0010	0	0
64	01110		1010101		1001		

Annex R
(informative)

Useful process control techniques

This Annex describes tools and procedures useful for monitoring and controlling the process of creating scannable Data Matrix symbols. These techniques do not constitute a print quality check of the produced symbols (the method of Clause 8 and Annex M is the required method for assessing symbol print quality) but they individually and collectively yield good indications of whether the symbol print process is creating workable symbols.

R.1 Symbol contrast

Most linear bar code verifiers have either a reflectometer mode or a mode for plotting scan reflectance profiles and/or reporting Symbol Contrast (as defined in ISO/IEC 15415 and ISO/IEC 19762) from undecodable scans. Except with symbols requiring special illumination configurations, the symbol contrast readings that can be obtained using a 6 or 10 mil aperture at 660 nm wavelength (either the reported SC value, the peak-to-peak scan profile excursions, or the difference between peak reflectometer readings) are found to correlate well with an image-derived symbol contrast. In particular these readings can be used to check that symbol contrast stays well above the minimum allowed for the intended symbol quality grade.

R.2 Special reference symbol

For process control purposes, a 16 x 16 ECC 200 reference symbol can be printed which encodes the data “30Q324343430794<OQQ”. As shown in Figure R.1, this reference symbol has a region of parallel bars and spaces which can be linearly scanned and then evaluated for print growth using the edge-measurement methodologies of ISO/IEC 15416.

Figure R.1 — ECC 200 reference symbol encoding "30Q324343430794<OQQ"

Many linear bar code verifiers can be programmed to list element widths derived by the ISO/IEC 15416 methodology even for undecoded scans. The left-hand portion of any linear test scan across the upper half of the ECC 200 reference symbol will contain four bar-space pairs whose widths may be designated b_1 to b_4 and s_1 to s_4.

A normalised indication of horizontal print growth can be calculated as:

$(b_1 + b_2 + b_3 + b_4) / (b_1 + s_1 + b_2 + s_2 + b_3 + s_3 + b_4 + s_4)$

This value in Data Matrix symbols should nominally be 50% and stay within 35% to 65% limits.

Note that this measurement will not be sensitive to printing variations parallel to the long dimension of the elements in the reference symbol. If a more complete assessment of the print process is desired, the Data Matrix reference symbol should be printed in both orientations and tested.

R.3 Assessing Axial Nonuniformity

For any symbol, measure the length of both legs of the "L" shaped finder pattern. Divide each length by the number of modules in that dimension, e.g. a 12 x 36 symbol would have 12 and 36 as divisors. These two normalised dimensions are X_{AVG} and Y_{AVG} which can be used in the formula below to grade Axial Nonuniformity.

$$AN = \text{abs}(X_{AVG} - Y_{AVG}) / ((X_{AVG} + Y_{AVG}) / 2)$$

If the value of AN is greater than 0,12 the symbol would fail according to ISO/IEC 15415. Values up to 0,06 correspond to a grade 4 for this parameter.

R.4 Visual inspection for symbol distortion and defects

Ongoing visual inspection of the perimeter patterns in sample symbols can monitor two important aspects of the print process. First, 2D matrix symbols are susceptible to errors caused by local distortions of the matrix grid. Any such distortions will show up visually in a Data Matrix symbol as either crooked edges on the "L" shaped finder pattern or uneven spacings within the alternating patterns found along the other two margins of the symbol. Larger ECC 200 symbols also include alignment patterns whose straightness and evenness can be visually checked. Symbols likely to fail the reference decode can be quickly identified in this way. Second, the two arms of the finder pattern and the adjacent quiet zones should always be solidly in opposite reflectance states. Failures in the print mechanism which may produce defects in the form of light or dark streaks through the symbol should be visibly evident where they infringe the finder or quiet zone. Such systematic failures in the print process should be corrected.

Annex S
(informative)

Autodiscrimination capability

Data Matrix may be read by suitably programmed bar code decoders which have been designed to autodiscriminate it from other symbologies. The decoder's valid set of symbologies should be limited to those needed by a given application to maximise reading security.

Annex T
(informative)

System considerations

Any Data Matrix application must be viewed as a total system solution. All the symbology encoding/decoding components (surface marker or printer, labels, readers) making up an application need to operate together as a system. A failure in any link of the chain, or a mismatch between them, could compromise the performance of the overall system.

— While compliance with the specifications is one key to assuring overall system success, other considerations come into play which may influence performance as well. The following guidelines suggest some factors to keep in mind when specifying or implementing bar code systems:

— Select a print density which will yield tolerance values that can be achieved by the marking or printing technology being used.

— Choose a reader with a resolution suitable for the symbol density and quality produced by the printing technology.

— Ensure that the printed symbol's optical properties are compatible with the wavelength of the scanner's light source or sensor.

— Verify symbol compliance in the final label or package configuration. Overlays, showthrough, and curved or irregular surfaces can all affect symbol readability.

Marking technologies that are not consistently capable of producing a solid line of continuous modules, for example, dot peen and ink jet, require particular care to ensure that gaps between nominally touching modules do not interfere with the decoding of the symbol using the application specified aperture size. In addition, the relative positioning of modules and the horizontal and vertical axes needs to comply with the requirements for axial non-uniformity specified in ISO/IEC 15415. Application specifications should also consult ISO/IEC 15415 for guidance regarding the specification of aperture size, lighting, and other parameters.

Scanning systems must take into consideration the variations in diffuse reflection between dark and light features. At some scanning angles, the specular component of the reflected light can greatly exceed the desired diffuse component, making successful reading more difficult. In cases where the surface of the part or material can be altered, matte, non-glossy finishes may help minimise specular effects. Where this option is not available, particular care must be taken to ensure the illumination for the mark being read optimises the desired contrast components.

Bibliography

[1] LIN and COSTELLO, "Error Control Coding: Fundamentals and Applications," Prentice Hall, 1983.

[2] C. BRITTON RORBAUGH, "Error Coding Cookbook," McGraw Hill, 1996.

[3] AIM Inc. "Data Matrix Developer's Diskette" (AIM Inc., 125 Warrendale-Bayne Road, Suite 100, Warrendale, PA 15086, USA).

blank

BSI - British Standards Institution

BSI is the independent national body responsible for preparing British Standards. It presents the UK view on standards in Europe and at the international level. It is incorporated by Royal Charter.

Revisions

British Standards are updated by amendment or revision. Users of British Standards should make sure that they possess the latest amendments or editions.

It is the constant aim of BSI to improve the quality of our products and services. We would be grateful if anyone finding an inaccuracy or ambiguity while using this British Standard would inform the Secretary of the technical committee responsible, the identity of which can be found on the inside front cover. Tel: +44 (0)20 8996 9000. Fax: +44 (0)20 8996 7400.

BSI offers members an individual updating service called PLUS which ensures that subscribers automatically receive the latest editions of standards.

Buying standards

Orders for all BSI, international and foreign standards publications should be addressed to Customer Services. Tel: +44 (0)20 8996 9001. Fax: +44 (0)20 8996 7001 Email: orders@bsigroup.com You may also buy directly using a debit/credit card from the BSI Shop on the Website http://www.bsigroup.com/shop

In response to orders for international standards, it is BSI policy to supply the BSI implementation of those that have been published as British Standards, unless otherwise requested.

Information on standards

BSI provides a wide range of information on national, European and international standards through its Library and its Technical Help to Exporters Service. Various BSI electronic information services are also available which give details on all its products and services. Contact Information Centre. Tel: +44 (0)20 8996 7111 Fax: +44 (0)20 8996 7048 Email: info@bsigroup.com

Subscribing members of BSI are kept up to date with standards developments and receive substantial discounts on the purchase price of standards. For details of these and other benefits contact Membership Administration. Tel: +44 (0)20 8996 7002 Fax: +44 (0)20 8996 7001 Email: membership@bsigroup.com

Information regarding online access to British Standards via British Standards Online can be found at http://www.bsigroup.com/BSOL

Further information about BSI is available on the BSI website at http://www.bsigroup.com

Copyright

Details and advice can be obtained from the Copyright and Licensing Manager. Tel: +44 (0)20 8996 7070 Email: copyright@bsigroup.com

BSI Group Headquarters 389 Chiswick High Road, London, W4 4AL, UK
Tel +44 (0)20 8996 9001
Fax +44 (0)20 8996 7001
www.bsigroup.com/standards

BS ISO/IEC 18004:2015

BSI Standards Publication

Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification

EXHIBIT
1002

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of ISO/IEC 18004:2015. It supersedes BS ISO/IEC 18004:2006 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/34, Automatic identification and data capture techniques.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

ISBN 978 0 580 81570 6

ICS 01.080.50; 35.040

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2015.

Amendments issued since publication

Date	Text affected

BS ISO/IEC 18004:2015

INTERNATIONAL STANDARD

ISO/IEC 18004

Third edition
2015-02-01

Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification

Technologies de l'information — Technologie d'identification automatique et de capture des données — Spécification de la symbologie de code à barres Code QR

Reference number
ISO/IEC 18004:2015(E)

Contents

Page

Foreword vi
Introduction vii
1 Scope 1
2 Conformance 1
3 Normative references 1
4 Terms and definitions 2
5 Mathematical and logical symbols, abbreviations and conventions 4
5.1 Mathematical and logical symbols 4
5.2 Abbreviations 4
5.3 Conventions 4
5.3.1 Module positions 4
5.3.2 Byte notation 4
5.3.3 Version references 4
6 Symbol description 4
6.1 Basic characteristics 4
6.2 Summary of additional features 6
6.3 Symbol structure 7
6.3.1 General 7
6.3.2 Symbol Versions and sizes 9
6.3.3 Finder pattern 16
6.3.4 Separator 17
6.3.5 Timing pattern 17
6.3.6 Alignment patterns 17
6.3.7 Encoding region 17
6.3.8 Quiet zone 17
7 Requirements 18
7.1 Encode procedure overview 18
7.2 Data analysis 20
7.3 Modes 20
7.3.1 General 20
7.3.2 Extended Channel Interpretation (ECI) mode 20
7.3.3 Numeric mode 21
7.3.4 Alphanumeric mode 21
7.3.5 Byte mode 21
7.3.6 Kanji mode 21
7.3.7 Mixing modes 21
7.3.8 Structured Append mode 21
7.3.9 FNC1 mode 22
7.4 Data encoding 22
7.4.1 Sequence of data 22
7.4.2 Extended Channel Interpretation (ECI) mode 23
7.4.3 Numeric mode 25
7.4.4 Alphanumeric mode 26
7.4.5 Byte mode 27
7.4.6 Kanji mode 29
7.4.7 Mixing modes 30
7.4.8 FNC1 modes 30
7.4.9 Terminator 32
7.4.10 Bit stream to codeword conversion 32
7.5 Error correction 36
7.5.1 Error correction capacity 36
7.5.2 Generating the error correction codewords 44

7.6 Constructing the final message codeword sequence 45
7.7 Codeword placement in matrix 46
7.7.1 Symbol character representation 46
7.7.2 Function pattern placement 46
7.7.3 Symbol character placement 46
7.8 Data masking 50
7.8.1 General 50
7.8.2 Data mask patterns 50
7.8.3 Evaluation of data masking results 53
7.9 Format information 55
7.9.1 QR Code symbols 55
7.9.2 Micro QR Code symbols 57
7.10 Version information 58

8 Structured Append 59
8.1 Basic principles 59
8.2 Symbol Sequence Indicator 60
8.3 Parity Data 61

9 Symbol printing and marking 61
9.1 Dimensions 61
9.2 Human-readable interpretation 61
9.3 Marking guidelines 61

10 Symbol quality 62
10.1 Methodology 62
10.2 Symbol quality parameters 62
10.2.1 Fixed pattern damage 62
10.2.2 Scan grade and overall symbol grade 62
10.2.3 Grid non-uniformity 62
10.3 Process control measurements 62

11 Decoding procedure overview 62

12 Reference decode algorithm for QR Code 63

13 Autodiscrimination capability 70

14 Transmitted data 70
14.1 General principles 70
14.2 Symbology Identifier 71
14.3 Extended Channel Interpretations 71
14.4 FNC1 72

Annex A (normative) **Error detection and correction generator polynomials 73**

Annex B (normative) **Error correction decoding steps 77**

Annex C (normative) **Format information 79**

Annex D (normative) **Version information 81**

Annex E (normative) **Position of alignment patterns 83**

Annex F (normative) **Symbology Identifier 85**

Annex G (normative) **QR Code print quality symbology-specific aspects 86**

Annex H (informative) **JIS8 and Shift JIS character sets 92**

Annex I (informative) **Symbol encoding examples 94**

Annex J (informative) **Optimisation of bit stream length 99**

Annex K (informative) **User guidelines for printing and scanning of QR Code symbols 108**

Annex L (informative) **Autodiscrimination 110**

Annex M (informative) **Process control techniques** **111**
Annex N (informative) **Characteristics of Model 1 symbols** **113**
Bibliography **116**

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 31, *Automatic identification and data capture techniques*.

This third edition cancels and replaces the second edition (ISO/IEC 18004:2006), which has been technically revised.

Introduction

It is necessary to distinguish four technically different, but closely related members of the QR Code family, which represent an evolutionary sequence.

— QR Code Model 1 was the original specification for QR Code and is described in AIM ITS 97-001 International Symbology Specification-QR Code.

— QR Code Model 2 was an enhanced form of the symbology with additional features (primarily the addition of alignment patterns to assist navigation in larger symbols), and was the basis of the first edition of ISO/IEC 18004.

— QR Code (the basis of the second edition of ISO/IEC 18004) is closely similar to QR Code Model 2, its QR Code format differs only in the addition of the facility for symbols to appear in a mirror image orientation for reflectance reversal (light symbols on dark backgrounds) and the option for specifying alternative character sets to the default.

— The Micro QR Code format (also specified in the second edition of ISO/IEC 18004), is a variant of QR Code with a reduced number of overhead modules and a restricted range of sizes, which enables small to moderate amount of data to be represented in a small symbol, particularly suited to direct marking on parts and components, and to applications where the space available for the symbol is severely restricted.

QR Code is a matrix symbology. The symbols consist of an array of nominally square modules arranged in an overall square pattern, including a unique finder pattern located at three corners of the symbol (in Micro QR Code symbols, at a single corner) and intended to assist in easy location of its position, size, and inclination. A wide range of sizes of symbol is provided for, together with four levels of error correction. Module dimensions are user-specified to enable symbol production by a wide variety of techniques.

QR Code Model 2 symbols are fully compatible with QR Code reading systems.

Model 1 QR Code symbols are recommended only to be used in closed system applications and it is not a requirement that equipment complying with this International Standard should support Model 1. Since QR Code is the recommended model for new, open systems application of QR Code, this International Standard describes QR Code fully, and lists the features in which Model 1 QR Code differs from QR Code in Annex N.

Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification

1 Scope

This International Standard defines the requirements for the symbology known as QR Code. It specifies the QR Code symbology characteristics, data character encoding methods, symbol formats, dimensional characteristics, error correction rules, reference decoding algorithm, production quality requirements, and user-selectable application parameters.

2 Conformance

QR Code symbols (and equipment designed to produce or read QR Code symbols) shall be considered as conforming with this International Standard if they provide or support the features defined in this International Standard.

Symbols complying with the requirements for QR Code Model 1, as described in ISO/IEC 18004:2006, may not be readable with equipment complying with this International Standard.

Symbols complying with the requirements for QR Code Model 2, as defined in ISO/IEC 18004:2000, are readable with equipment complying with this International Standard.

Reading equipment complying with ISO/IEC 18004:2000 will not be able to read all symbols complying with this International Standard. Symbols that make use of the additional features of QR Code will not be readable by such equipment.

Printing equipment complying with ISO/IEC 18004:2000 will not be able to print all symbols defined in this International Standard. Symbols that make use of the additional features of QR Code will not be printable by such equipment.

It should be noted, however, that QR Code Model 2 and Micro QR Code are the form of the symbology recommended for new and open systems applications.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8859-1:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

ISO/IEC 15415, *Information technology — Automatic identification and data capture techniques — Bar code symbol print quality test specification — Two-dimensional symbols*

ISO/IEC 19762-1, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 1: General terms relating to AIDC*

ISO/IEC 19762-2, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 2: Optically readable media (ORM)*

JIS X 0201, 7-bit and 8-bit coded character sets for information interchange

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762-1 and ISO/IEC 19762-2 and the following apply.

4.1
character count indicator
bit sequence which defines the data string length in a mode

4.2
data masking
process of XORing the bit pattern in the encoding region with a data mask pattern to provide a symbol with more evenly balanced numbers of dark and light modules, and reduced occurrence of patterns which would interfere with fast processing of the image

4.3
data mask pattern reference
three-bit identifier of the data masking patterns applied to the symbol

4.4
encoding region
region of the symbol not occupied by function patterns and available for encoding of data and error correction codewords, and for Version and format information

4.5
exclusive subset
subset of characters within the character set of a mode which are not shared with the more restricted character set of another mode

4.6
extension pattern
function pattern in Model 1 symbols, which does not encode data

4.7
format information
encoded pattern containing information on symbol characteristics essential to enable the remainder of the encoding region to be decoded

4.8
QR Code
pertaining to QR Code symbols identified as versions 1 to 40, as distinct from Micro QR Code symbols

4.9
function pattern
overhead component of the symbol (finder, separator, timing patterns, and alignment patterns) required for location of the symbol or identification of its characteristics to assist in decoding

4.10
masking
process of XORing the bit pattern in an area of the symbol with a mask pattern to reduce the occurrence of patterns which would interfere with fast processing of the image

4.11
micro
pertaining to Micro QR Code symbols identified as versions M1 to M4, as distinct from QR Code symbols

4.12
mode
method of representing a defined character set as a bit string

4.13
mode indicator
four-bit identifier indicating in which mode the following data sequence is encoded

4.14
padding bit
zero bit, not representing data, used to fill empty positions of the final codeword after the terminator in a data bit string

4.15
remainder bit
zero bit, not representing data, used to fill empty positions of the symbol encoding region after the final symbol character, where the area of the encoding region available for symbol characters does not divide exactly into 8-bit symbol characters

4.16
remainder codeword
pad codeword, placed after the error correction codewords, used to fill empty codeword positions to complete the symbol if the total number of data and error correction codewords does not exactly fill its nominal capacity

4.17
segment
sequence of data encoded according to the rules of one ECI or encoding mode

4.18
separator
function pattern of all light modules, one module wide, separating the finder patterns from the rest of the symbol

4.19
symbol number
three-bit field indicating the symbol version and error correction level applied, used as part of the format information in Micro QR Code symbols

4.20
terminator
bit pattern of defined number (depending on symbol) of all zero bits used to end the bit string representing data

4.21
timing pattern
alternating sequence of dark and light modules enabling module coordinates in the symbol to be determined

4.22
version
size of the symbol represented in terms of its position in the sequence of permissible sizes for Micro QR Code symbols from 11 × 11 modules (version M1) to 17 × 17 modules (version M4) or, for QR Code symbols, from 21 × 21 modules (version 1) to 177 × 177 (version 40) modules

Note 1 to entry: The error correction level applied to the symbol may be suffixed to the version designation, e.g. version 4-L or version M3-Q.

4.23
version information
encoded pattern in certain QR Code symbols containing information on the symbol version together with error correction bits for this data

5 Mathematical and logical symbols, abbreviations and conventions

5.1 Mathematical and logical symbols

Mathematical symbols used in formulae and equations are defined after the formula or equation in which they appear.

For the purposes of this document, the following mathematical operations apply.

div is the integer division operator;

mod is the integer remainder after division;

XOR is the exclusive-or logic function whose output is one only when its two inputs are not equivalent. It is represented by the symbol $\oplus$.

5.2 Abbreviations

BCH Bose-Chaudhuri-Hocquenghem

ECI Extended Channel Interpretation

RS Reed-Solomon

5.3 Conventions

5.3.1 Module positions

For ease of reference, module positions are defined by their row and column coordinates in the symbol, in the form (i, j) where i designates the row (counting from the top downwards) and j the column (counting from left to right) in which the module is located, with counting commencing at 0. Module (0, 0) is therefore located at the upper left corner of the symbol.

5.3.2 Byte notation

Byte contents are shown as hex values.

5.3.3 Version references

For QR Code symbols, symbol versions are referred to in the form Version V-E where V identifies the version number (1 to 40) and E indicates the error correction level (L, M, Q, H).

For Micro QR Code symbols, symbol versions are referred to in the form Version MV-E where the letter M indicates the Micro QR Code format and V (with a range of 1 to 4) and E (with values L, M and Q) have the meanings defined above.

6 Symbol description

6.1 Basic characteristics

QR Code is a matrix symbology with the following characteristics:

a) Formats:

1) QR Code, with full range of capabilities and maximum data capacity;

2) Micro QR Code, with reduced overhead, some restrictions on capabilities and reduced data capacity (compared with QR Code symbols).

b) Encodable character set:

1) numeric data (digits 0 - 9);

2) alphanumeric data (digits 0 - 9; upper case letters A - Z; nine other characters: space, $ % * + - . / :);

3) byte data [default: ISO/IEC 8859-1; or other sets as otherwise defined (see 7.3.5)];

4) Kanji characters. Kanji characters in QR Code can be compacted into 13 bits.

c) Representation of data:

A dark module is nominally a binary one and a light module is nominally a binary zero. However, see 6.2 for details of reflectance reversal.

d) Symbol size (not including quiet zone):

1) Micro QR Code symbols: 11 × 11 modules to 17 × 17 modules (Versions M1 to M4, increasing in steps of two modules per side);

2) QR Code symbols: 21 × 21 modules to 177 × 177 modules (Versions 1 to 40, increasing in steps of four modules per side).

e) Data characters per symbol

1) maximum Micro QR Code symbol size, Version M4-L):

— numeric data: 35 characters

— alphanumeric data: 21 characters

— Byte data: 15 characters

— Kanji data: 9 characters

2) maximum QR Code symbol size, Version 40-L:

— numeric data: 7 089 characters

— alphanumeric data: 4 296 characters

— Byte data: 2 953 characters

— Kanji data: 1 817 characters

f) Selectable error correction:

Four levels of Reed-Solomon error correction (referred to as L, M, Q and H in increasing order of capacity) allowing recovery of:

— L 7%

— M 15%

— Q 25%

— H 30%

of the symbol codewords.

For Micro QR Code symbols, error correction level H is not available. For Version M1 Micro QR Code symbols, the RS capacity is limited to error detection only.

g) Code type:

Matrix

h) Orientation independence:

Yes (both rotation and reflection)

Figure 1 illustrates a Version 1 QR Code symbol in normal colour and with reflectance reversal (see 6.2), in both normal and mirror image orientations.

Figure 2 illustrates a Version M2 Micro QR Code symbol in normal colour and with reflectance reversal (see 6.2), in both normal and mirror image orientations.

6.2 Summary of additional features

The use of the following additional features is optional in QR Code:

— Structured append

This allows files of data to be represented logically and continuously in up to 16 QR Code symbols. These may be scanned in any sequence to enable the original data to be correctly reconstructed. Structured Append is not available with Micro QR Code symbols.

— Extended Channel Interpretations

This mechanism enables data using character sets other than the default encodable set (e.g. Arabic, Cyrillic, Greek) and other data interpretations (e.g. compacted data using defined compression schemes) or other industry-specific requirements to be encoded. Extended Channel Interpretations other than the default interpretation are not available in Micro QR Code symbols.

— Reflectance reversal

Symbols are intended to be read when marked so that the image is either dark on light or light on dark (see Figures 1 and 2). The specifications in this International Standard are based on dark images on a light background, therefore in the case of symbols produced with reflectance reversal references to dark or light modules should be taken as references to light or dark modules respectively.

— Mirror imaging

The arrangement of modules defined in this International Standard represents the “normal” orientation of the symbol. It is, however, possible to achieve a valid decode of a symbol in which the arrangement of the modules has been laterally transposed. When viewed with the finder patterns at the top left, top right and bottom left corners of the symbol, the effect of mirror imaging is to interchange the row and column positions of the modules.

Figure 1 — Examples of QR Code symbol encoding the text "QR Code Symbol" – (a) normal orientation and normal reflectance arrangement; (b) normal orientation and reversed reflectances; (c) mirror image orientation and normal reflectance arrangement; (d) mirror image orientation and reversed reflectances

Figure 2 — Examples of Version M2 Micro QR Code symbol encoding the text "01234567" – (a) normal orientation and normal reflectance arrangement; (b) normal orientation and reversed reflectances; (c) mirror image orientation and normal reflectance arrangement; (d) mirror image orientation and reversed reflectances

NOTE The corner marks in Figures 1 and 2 indicate the extent of the quiet zone.

6.3 Symbol structure

6.3.1 General

Each QR Code symbol shall be constructed of nominally square modules set out in a regular square array and shall consist of an encoding region and function patterns, namely finder, separator, timing patterns,

and alignment patterns. Function patterns do not encode data. The symbol shall be surrounded on all four sides by a quiet zone border. Figure 3 illustrates the structure of a Version 7 symbol. Figure 4 illustrates the structure of a Version M3 symbol.

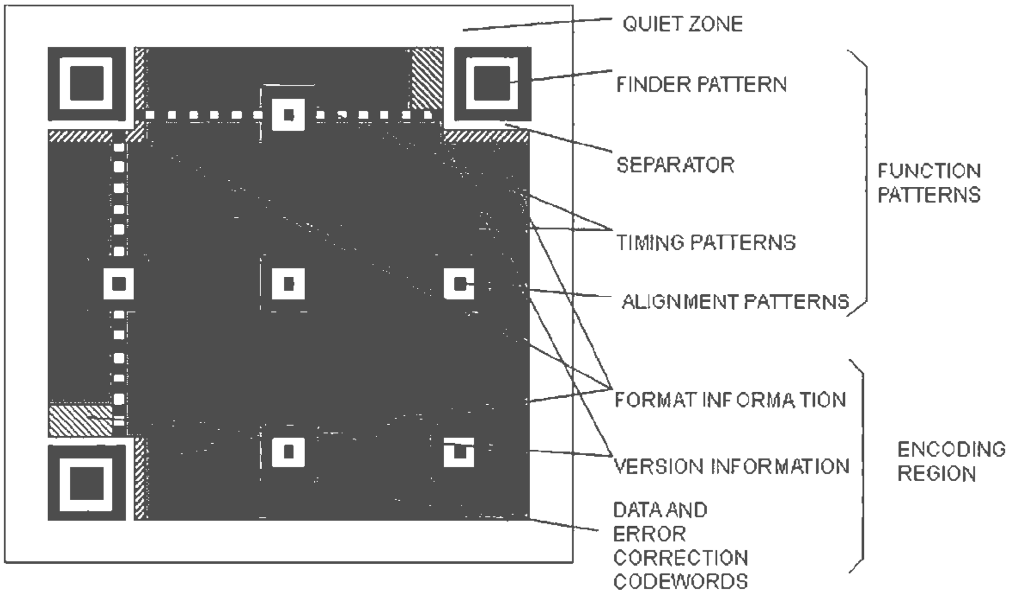


Figure 3 — Structure of a QR Code symbol

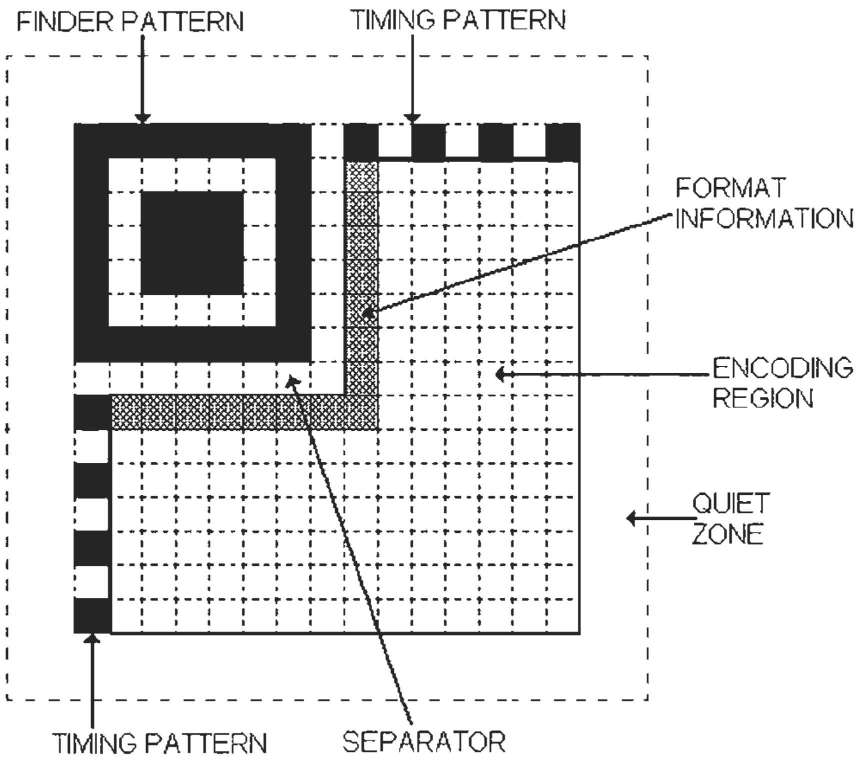


Figure 4 — Structure of Version M3 Micro QR Code symbol

6.3.2 Symbol Versions and sizes

6.3.2.1 QR Code symbols

There are forty sizes of QR Code symbol referred to as Version 1, Version 2 ... Version 40. Version 1 measures 21 modules × 21 modules, Version 2 measures 25 modules × 25 modules and so on increasing in steps of 4 modules per side up to Version 40 which measures 177 modules × 177 modules. Figures 5 to 10 illustrate the structure of Versions 1, 2, 6, 7, 14, 21 and 40.

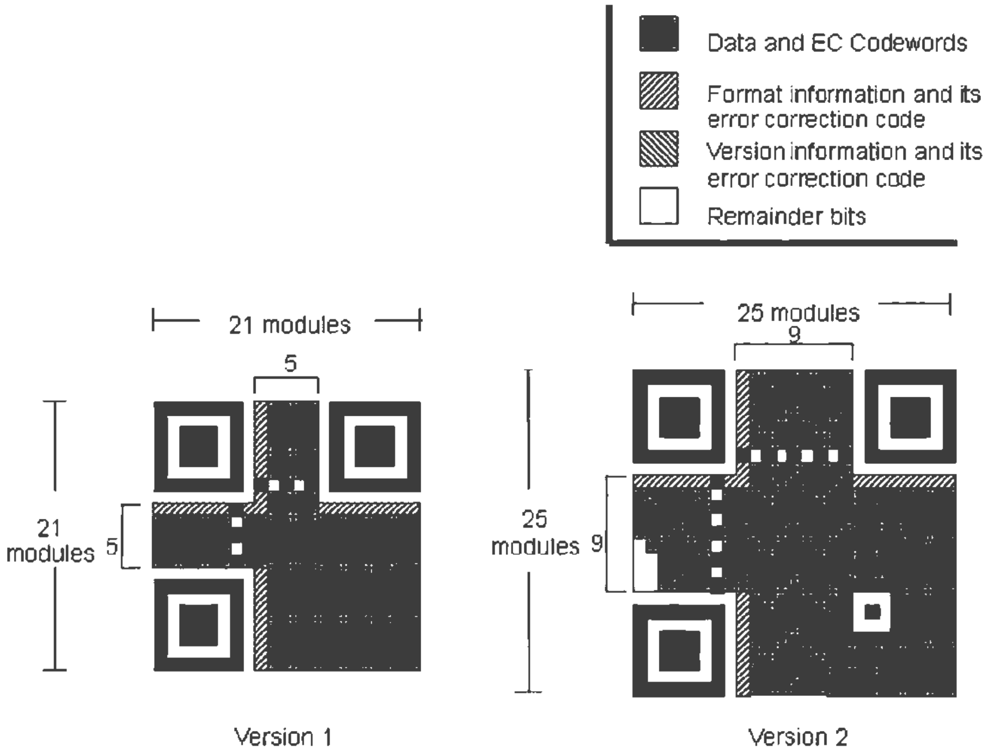


Figure 5 — Version 1 and 2 symbols

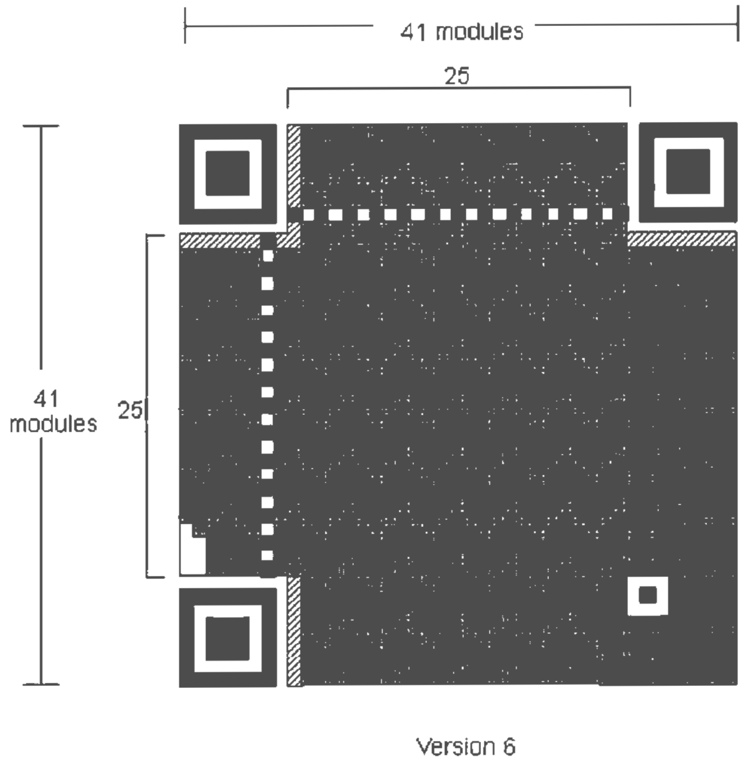


Figure 6 — Version 6 symbol

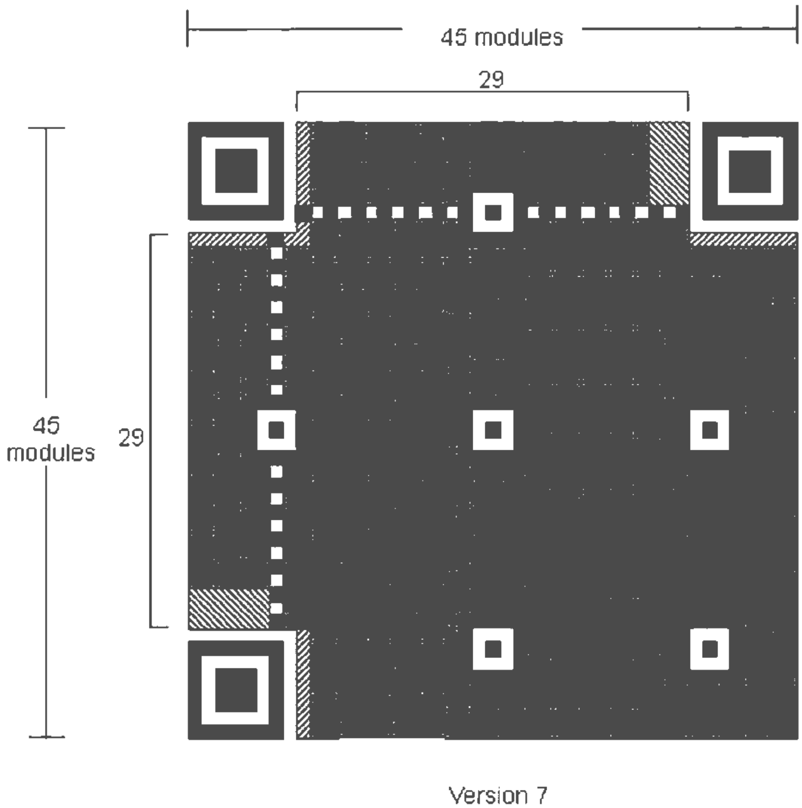


Figure 7 — Version 7 symbol

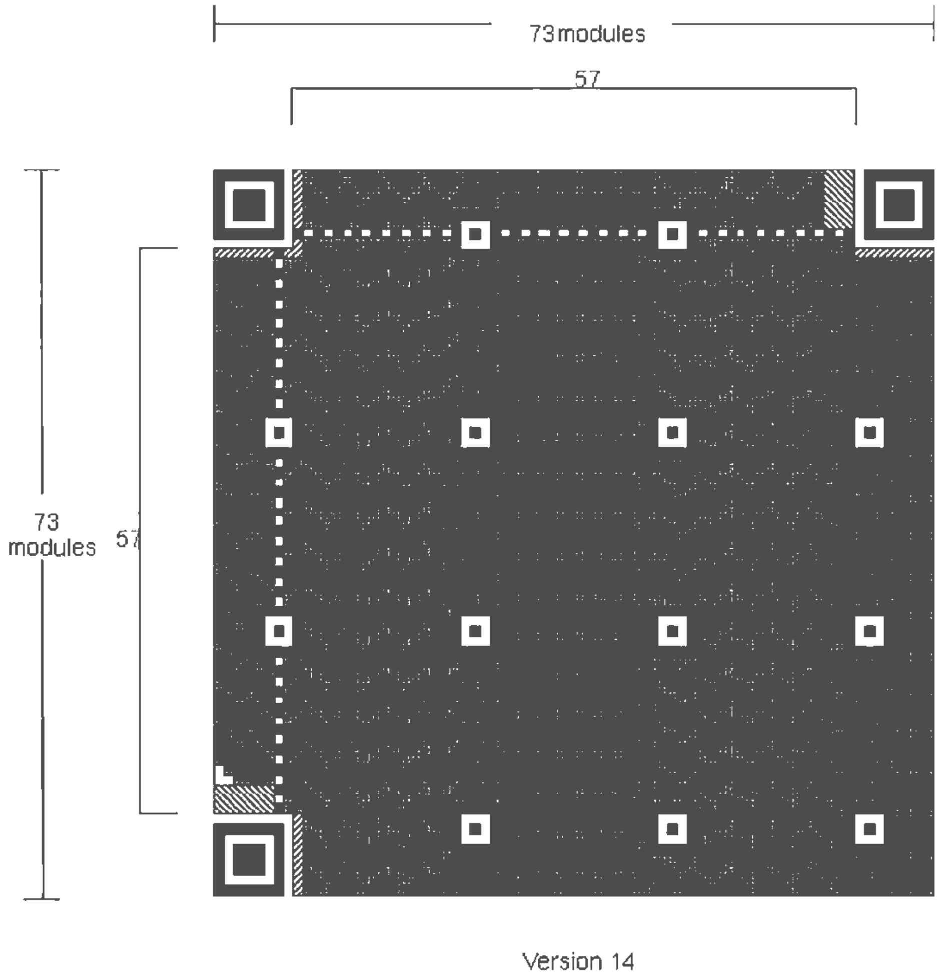


Figure 8 — Version 14 symbol

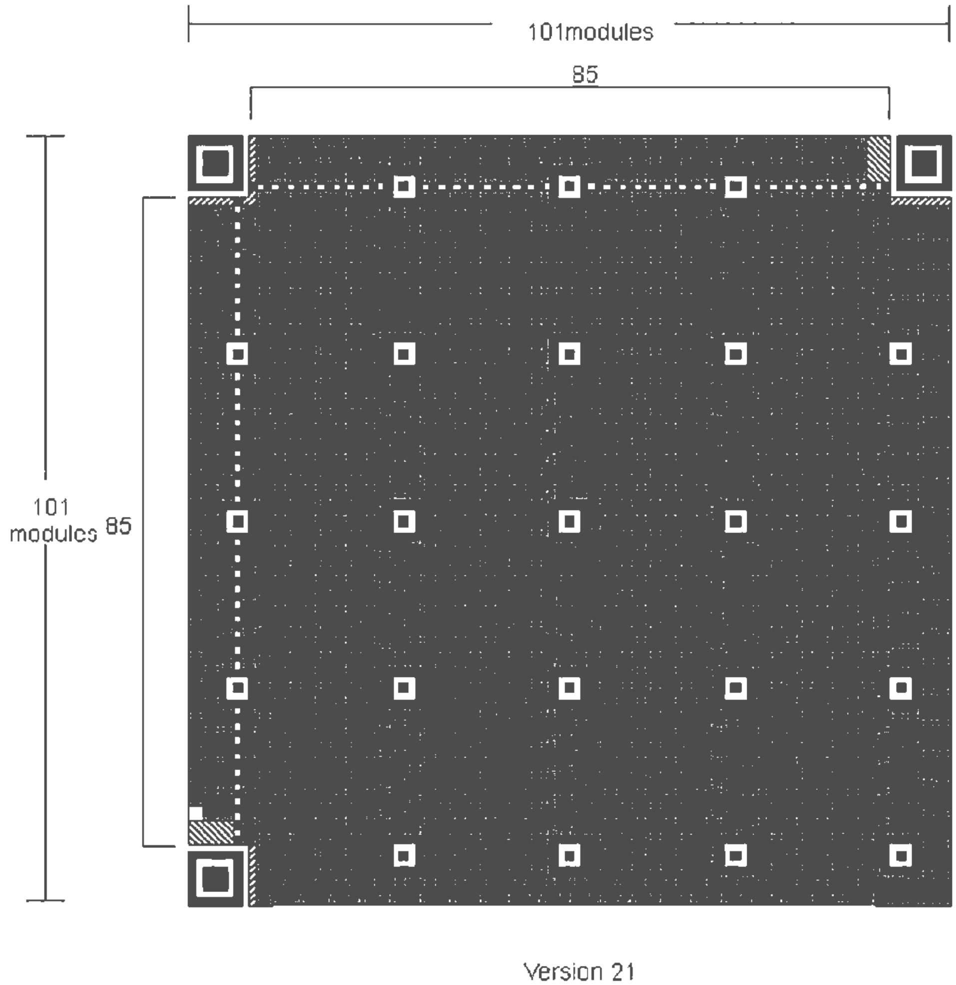


Figure 9 — Version 21 symbol

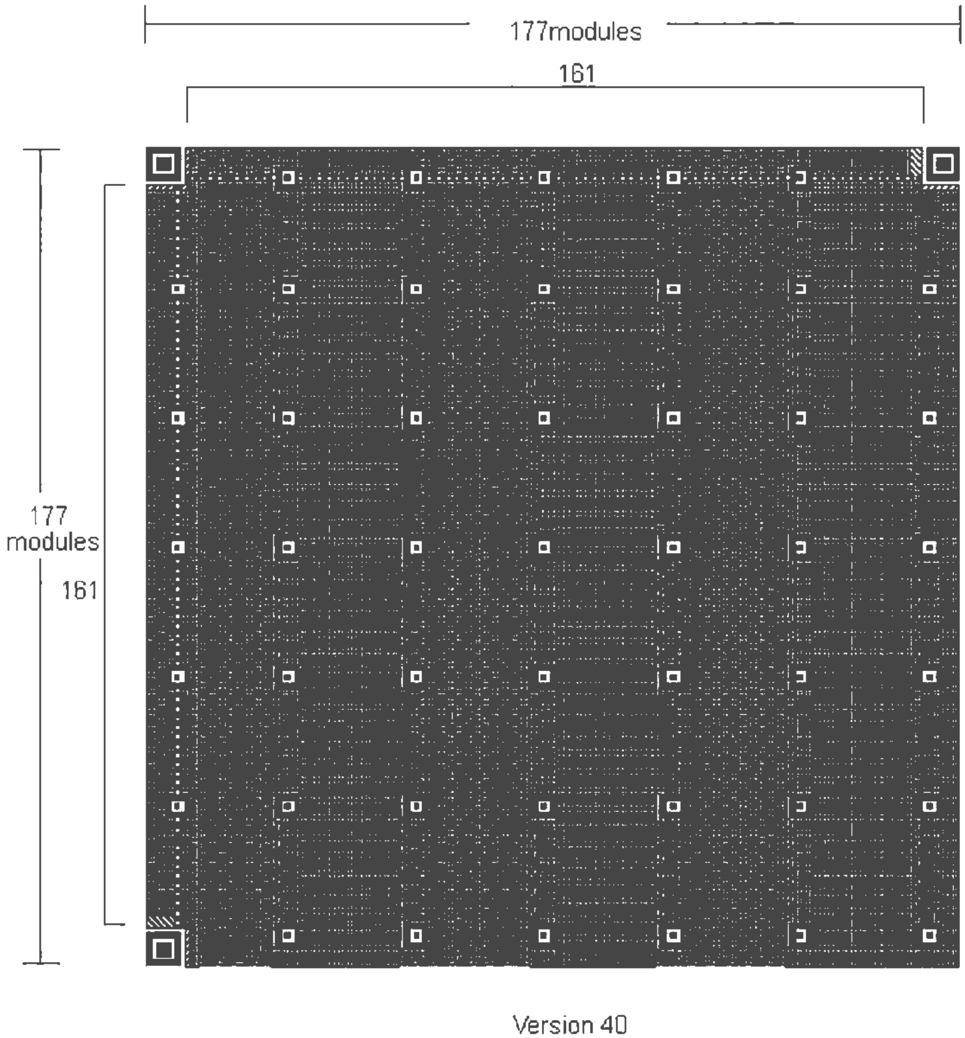


Figure 10 — Version 40 symbol

6.3.2.2 Micro QR Code symbols

There are four sizes of Micro QR Code symbol, referred to as Versions M1 to M4. Version M1 measures 11 × 11 modules, Version M2 13 × 13 modules, Version M3 15 × 15 modules, and Version M4 17 × 17 modules, i.e. increasing in steps of 2 modules per side. Figure 11 illustrates the structure of Micro QR Code Versions M1 to M4.

NOTE Two formats of M3 symbol are shown, which differ only in the codeword placement according to the error correction level.

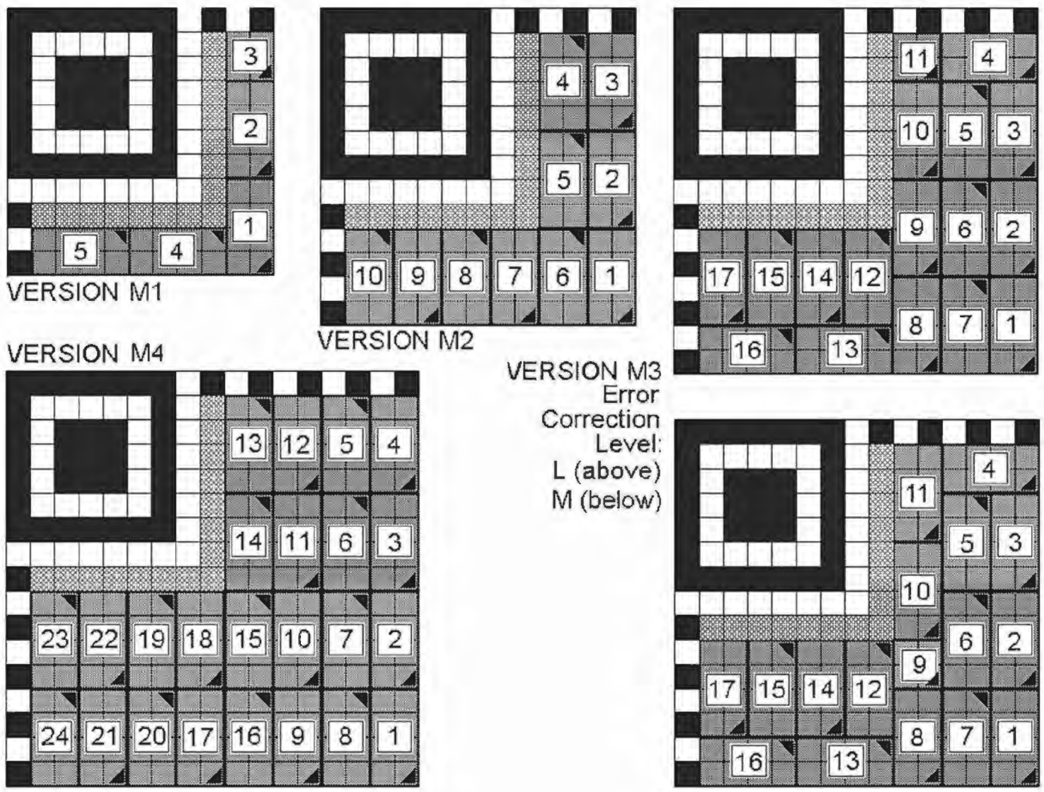


Figure 11 — Versions of Micro QR Code symbol

6.3.3 Finder pattern

6.3.3.1 QR Code symbols

There are three identical Finder Patterns located at the upper left, upper right and lower left corners of the symbol respectively as illustrated in Figure 3. Each finder pattern may be viewed as three superimposed concentric squares and is constructed of dark 7 × 7 modules, light 5 × 5 modules and dark 3 × 3 modules. The ratio of module widths in each finder pattern is 1 : 1 : 3 : 1 : 1 as illustrated in Figure 12. The symbol is preferentially encoded so that similar patterns have a low probability of being encountered elsewhere in the symbol, enabling rapid identification of a possible QR Code symbol in the field of view. Identification of the three finder patterns comprising the Finder Pattern then unambiguously defines the location and rotational orientation of the symbol in the field of view.

6.3.3.2 Micro QR Code symbols

A single finder pattern, as defined in 6.3.3.1, is located at the upper left corner of the symbol as illustrated in Figure 4. Identification of the finder pattern together with the timing patterns unambiguously defines the size, location and rotational orientation of the symbol in the field of view.

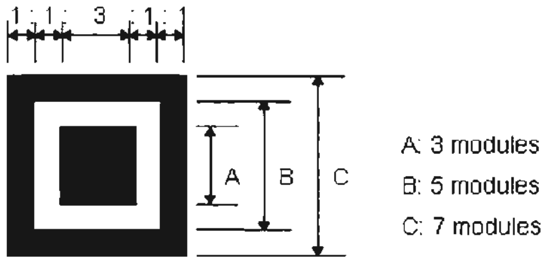


Figure 12 — Structure of finder pattern

6.3.4 Separator

A one-module wide separator, constructed of all light modules, is placed between each finder pattern and the Encoding Region, as illustrated in Figures 3 and 4.

6.3.5 Timing pattern

The horizontal and vertical timing patterns respectively consist of a one module wide row or column of alternating dark and light modules, commencing and ending with a dark module. They enable the symbol density and version to be determined and provide datum positions for determining module coordinates.

In QR Code symbols, the horizontal timing pattern runs across row 6 of the symbol between the separators for the upper finder patterns; the vertical timing pattern similarly runs down column 6 of the symbol between the separators for the left-hand finder patterns. See Figure 3.

In Micro QR Code symbols, the horizontal timing pattern runs across row 0 of the symbol on the right side of the separator to the right hand edge of the symbol; the vertical Timing Pattern similarly runs down column 0 of the symbol below the separator to the bottom edge of the symbol. See Figure 4.

6.3.6 Alignment patterns

Alignment patterns are present only in QR Code symbols of version 2 or larger. Each alignment pattern may be viewed as three superimposed concentric squares and is constructed of dark 5 × 5 modules, light 3 × 3 modules and a single central dark module. The number of alignment patterns depends on the symbol version and they shall be placed in all symbols of Version 2 or larger in positions defined in Annex E.

6.3.7 Encoding region

This region shall contain the symbol characters representing data, those representing error correction codewords, the format information and, where appropriate, the version information. Refer to 7.7.1 for details of the symbol characters. Refer to 7.9 for details of the format information. Refer to 7.10 for details of the version information.

6.3.8 Quiet zone

This is a region which shall be free of all other markings, surrounding the symbol on all four sides. Its nominal reflectance value shall be equal to that of the light modules.

For QR Code symbols its width shall be 4X.

For Micro QR Code symbols its width shall be 2X.

7 Requirements

7.1 Encode procedure overview

This section provides an overview of the steps required to convert input data to a QR Code symbol.

Step 1 Data analysis

Analyze the input data stream to identify the variety of different characters to be encoded. The QR Code format (but not the Micro QR Code format) supports the Extended Channel Interpretation feature, enabling data differing from the default character set to be encoded. QR Code includes several modes (see 7.3) to allow different sub-sets of characters to be converted into symbol characters in efficient ways. Switch between modes as necessary in order to achieve the most efficient conversion of data into a binary string. Select the required Error Detection and Correction Level. If the user has not specified the symbol version to be used, select the smallest version that will accommodate the data. A complete list of symbol versions and capacities is shown in Table 1.

Step 2 Data encoding

Convert the data characters into a bit stream in accordance with the rules for the mode in force, as defined in 7.4.2 to 7.4.6, inserting mode indicators as necessary to change modes at the beginning of each new mode segment, and a Terminator at the end of the data sequence. Split the resulting bit stream into 8-bit codewords. Add Pad Characters as necessary to fill the number of data codewords required for the version.

Step 3 Error correction coding

Divide the codeword sequence into the required number of blocks (as defined in Table 9) to enable the error correction algorithms to be processed. Generate the error correction codewords for each block, appending the error correction codewords to the end of the data codeword sequence.

Step 4 Structure final message

Interleave the data and error correction codewords from each block as described in 7.6 (step 3) and add remainder bits as necessary.

Step 5 Module placement in matrix

Place the codeword modules in the matrix together with the finder pattern, separators, timing pattern, and (if required) alignment patterns.

Step 6 Data masking

Apply the data masking patterns in turn to the encoding region of the symbol. Evaluate the results and select the pattern which optimizes the dark/light module balance and minimizes the occurrence of undesirable patterns.

Step 7 Format and version information

Generate the format information and (where applicable) the version information and complete the symbol.

Table 1 — Codeword capacity of all versions of QR Code

Version	No. of Modules/ side (A)	Function pattern modules (B)	Format and version information modules (C)	Data modules except (C) ($D = A^2 - B - C$)	Data capacity [codewords] [a] (E)	Remainder Bits
M1	11	70	15	36	5	0

[a] All codewords are 8 bits in length, except in versions M1 and M3 where the final data codeword is 4 bits in length

Table 1 *(continued)*

Version	No. of Modules/ side (A)	Function pattern modules (B)	Format and version information modules (C)	Data modules except (C) (D = A² – B –C)	Data capacity [codewords] [a] (E)	Remainder Bits
M2	13	74	15	80	10	0
M3	15	78	15	132	17	0
M4	17	82	15	192	24	0
1	21	202	31	208	26	0
2	25	235	31	359	44	7
3	29	243	31	567	70	7
4	33	251	31	807	100	7
5	37	259	31	1 079	134	7
6	41	267	31	1 383	172	7
7	45	390	67	1 568	196	0
8	49	398	67	1 936	242	0
9	53	406	67	2 336	292	0
10	57	414	67	2 768	346	0
11	61	422	67	3 232	404	0
12	65	430	67	3 728	466	0
13	69	438	67	4 256	532	0
14	73	611	67	4 651	581	3
15	77	619	67	5 243	655	3
16	81	627	67	5 867	733	3
17	85	635	67	6 523	815	3
18	89	643	67	7 211	901	3
19	93	651	67	7 931	991	3
20	97	659	67	8 683	1 085	3
21	101	882	67	9 252	1 156	4
22	105	890	67	10 068	1 258	4
23	109	898	67	10 916	1 364	4
24	113	906	67	11 796	1 474	4
25	117	914	67	12 708	1 588	4
26	121	922	67	13 652	1 706	4
27	125	930	67	14 628	1 828	4
28	129	1 203	67	15 371	1 921	3
29	133	1 211	67	16 411	2 051	3
30	137	1 219	67	17 483	2 185	3
31	141	1 227	67	18 587	2 323	3
32	145	1 235	67	19 723	2 465	3
33	149	1 243	67	20 891	2 611	3
34	153	1 251	67	22 091	2 761	3
35	157	1 574	67	23 008	2 876	0

[a] All codewords are 8 bits in length, except in versions M1 and M3 where the final data codeword is 4 bits in length

Table 1 *(continued)*

Version	No. of Modules/ side (A)	Function pattern modules (B)	Format and version information modules (C)	Data modules except (C) (D = A^2 – B – C)	Data capacity [codewords] [a] (E)	Remainder Bits
36	161	1 582	67	24 272	3 034	0
37	165	1 590	67	25 568	3 196	0
38	169	1 598	67	26 896	3 362	0
39	173	1 606	67	28 256	3 532	0
40	177	1 614	67	29 648	3 706	0

[a] All codewords are 8 bits in length, except in versions M1 and M3 where the final data codeword is 4 bits in length

7.2 Data analysis

Analyze the input data string to determine its content and select the default or other appropriate ECI and the appropriate mode to encode each sequence as described in 7.4. Each mode in sequence from Numeric mode to Kanji mode progressively requires more bits per character. It is possible to switch from mode to mode within a symbol in order to minimize the bit stream length for data, parts of which can more efficiently be encoded in one mode than other parts, e.g. numeric sequences followed by alphanumeric sequences. It is in theory most efficient to encode data in the mode requiring the fewest bits per data character, but as there is some overhead in the form of mode indicator and character count indicator associated with each mode change, it may not always result in the shortest overall bit stream to change modes for a small number of characters. Also, because the capacity of symbols increases in discrete steps from one version to the next, it may not always be necessary to achieve the maximum conversion efficiency in every case. Guidance on minimising the bit stream length is given in Annex J. In Micro QR Code symbols, there are restrictions on the modes available in the smaller versions. Annex J.2 shows the Micro QR Code symbol versions appropriate for various combinations of two modes.

7.3 Modes

7.3.1 General

The modes defined below are based on the character values and assignments associated with the default ECI. When any other ECI is in force (in QR Code symbols only), the byte values rather than the specific character assignments shall be used to select the optimum data compaction mode. For example, Numeric mode would be appropriate if there is a sequence of data byte values within the range 30_{HEX} to 39_{HEX} inclusive. In this case the compaction is carried out using the default numeric or alphabetic equivalents of the byte values.

7.3.2 Extended Channel Interpretation (ECI) mode

The Extended Channel Interpretation (ECI) protocol defined in the AIM Inc. International Technical Specification *Extended Channel Interpretations*, allows the output data stream to have interpretations different from that of the default character set. The ECI protocol is defined consistently across a number of symbologies. The ECI protocol provides a consistent method to specify particular interpretations of byte values before printing and after decoding. The ECI protocol is not supported in Micro QR Code symbols.

The default interpretation for QR Code is ECI **000003** representing the ISO/IEC 8859-1 character set.

International applications using other character sets should use the ECI protocol. For instance, the interpretation corresponding to the JIS8 and Shift JIS character sets is ECI **000020**.

The effect of ECI mode is to insert an ECI escape sequence at that point in the data. It is immediately followed by another mode indicator (e.g. for efficient data encoding) and remains in force until the end of the message or a subsequent ECI mode indicator.

7.3.3 Numeric mode

Numeric mode encodes data from the decimal digit set (**0 - 9**) (byte values 30_{HEX} to 39_{HEX}). Normally, 3 data characters are represented by 10 bits.

7.3.4 Alphanumeric mode

Alphanumeric mode encodes data from a set of 45 characters, i.e. 10 numeric digits (**0 - 9**) (byte values 30_{HEX} to 39_{HEX}), 26 alphabetic characters (**A - Z**) (byte values 41_{HEX} to $5A_{HEX}$), and 9 symbols (**SP, $, %, *, +, -, ., /, :**) (byte values 20_{HEX}, 24_{HEX}, 25_{HEX}, $2A_{HEX}$, $2B_{HEX}$, 2D to $2F_{HEX}$, $3A_{HEX}$ respectively). Normally, two input characters are represented by 11 bits.

Alphanumeric mode is not available in Version M1 Micro QR Code symbols.

7.3.5 Byte mode

In this mode, data is encoded at 8 bits per character.

In closed-system national or application-specific implementations of QR Code, an alternative 8-bit character set, for example as defined in an appropriate part of ISO/IEC 8859, may be specified for Byte mode. When an alternative character set is specified, however, the parties intending to read the QR Code symbols require to be notified of the applicable character set in the application specification or by bilateral agreement.

Byte mode is not available in Version M1 or M2 Micro QR Code symbols.

7.3.6 Kanji mode

The Kanji mode efficiently encodes Kanji characters in accordance with the Shift JIS system based on JIS X 0208. The Shift JIS values are shifted from the JIS X 0208 values. JIS X 0208 gives details of the shift coded representation. Each two-byte character value is compacted to a 13-bit binary codeword.

When the character set specified for 8-bit byte mode makes use of byte values in the ranges 81_{HEX} to $9F_{HEX}$ and/or $E0_{HEX}$ to EB_{HEX}, it may not be possible to use Kanji mode unambiguously, as reading systems will be unable to determine from the transmitted data whether such byte values are the lead byte of a double byte character. It may be possible to achieve a shorter bit stream by using the Kanji mode compaction rules when an appropriate sequence of byte values occurs in the data (i.e. lead bytes in the ranges 81_{HEX} to $9F_{HEX}$ and/or $E0_{HEX}$ to EB_{HEX} followed by trailer bytes in the range 40_{HEX} to FC_{HEX},except $7F_{HEX}$, or EB_{HEX} followed by 40_{HEX} to BF_{HEX}). Figure H.1 shows the byte combinations graphically.

Kanji mode is not available in version M1 or M2 Micro QR Code symbols.

7.3.7 Mixing modes

The QR Code symbol may contain sequences of data in a combination of any of the modes described in 7.3.2 to 7.3.9. Micro QR Code symbols may contain sequences of data in a combination of any of the modes available for the version of the symbol and described in 7.3.3 to 7.3.7.

Refer to Annex J for guidance on selecting the most efficient way of representing a given input data string in multiple modes in QR Code symbols, and to Annex J.3 for the available versions of Micro QR Code symbols for given combinations of data in two modes.

7.3.8 Structured Append mode

Structured Append mode is used to split the encoding of the data from a message over a number of QR Code symbols. All of the symbols require to be read and the data message can be reconstructed in the correct sequence. The Structured Append header is encoded in each symbol to identify the length of

the sequence and the symbol's position in it, and verify that all the symbols read belong to the same message. Refer to 8 for details of encoding in Structured Append mode.

Structured Append mode is not available for Micro QR Code symbols.

7.3.9 FNC1 mode

FNC1 mode is used for messages containing specific data formats. In the "1st position" it designates data formatted in accordance with the GS1 General Specifications. In the "2nd position" it designates data formatted in accordance with a specific industry application previously agreed with AIM Inc. FNC1 mode applies to the entire symbol and is not affected by subsequent mode indicators.

NOTE "1st position" and 2nd position" do not refer to actual locations but are based on the positions of the character in Code 128 symbols, when used in an equivalent manner.

FNC1 mode is not available for Micro QR Code symbols.

7.4 Data encoding

7.4.1 Sequence of data

Input data is converted into a bit stream consisting of one or more segments each in a separate mode. In the default ECI, the bit stream commences with the first mode indicator. If the initial ECI is other than the default ECI, the bit stream commences with an ECI header, followed by the first segment.

The ECI header (if present) shall comprise:

— ECI mode indicator (4 bits)

— ECI Designator (8, 16 or 24 bits)

The ECI header shall begin with the first (most significant) bit of the ECI mode indicator and end with the final (least significant) bit of the ECI Designator.

The remainder of the bit stream is then made up of segments each comprising:

— Mode indicator

— Character count indicator

— Data bit stream

Each mode segment shall begin with the first (most significant) bit of the mode indicator and end with the final (least significant) bit of the data bit stream. There shall be no explicit separator between segments as their length is defined unambiguously by the rules for the mode in force and the number of input data characters.

To encode a sequence of input data in a given mode, the steps defined in sections 7.4.2 to 7.4.7 shall be followed. Table 2 defines the mode indicators for each mode. Table 3 defines the length of the character count indicator, which varies according to the mode and the symbol version in use.

Table 2 — Mode indicators for QR Code

Mode	QR Code symbols	Micro QR Code symbols			
Version	**all**	**M1**	**M2**	**M3**	**M4**
Mode indicator length (bits)	**4**	0	1	2	3
ECI	**0111**	n/a	n/a	n/a	n/a
Numeric	**0001**	n/a	0	00	000
Alphanumeric	**0010**	n/a	1	01	001
Byte	**0100**	n/a	n/a	10	010
Kanji	**1000**	n/a	n/a	11	011
Structured Append	**0011**	n/a	n/a	n/a	n/a
FNC1 [a]	0101 (1st position) **1001** (2nd position)	n/a	n/a	n/a	n/a
Terminator (End of Message) [b]	**0000**	**000**	**00000**	**0000000**	**000000000**

[a] See 7.4.8.2 and 7.4.8.3.

[b] The Terminator is not a mode indicator as such.

Table 3 — Number of bits in character count indicator for QR Code

Version	Numeric mode	Alphanumeric mode	Byte mode	Kanji mode
M1	**3**	n/a	n/a	n/a
M2	**4**	**3**	n/a	n/a
M3	**5**	**4**	**4**	**3**
M4	**6**	**5**	**5**	**4**
1 to 9	**10**	**9**	**8**	**8**
10 to 26	**12**	**11**	**16**	**10**
27 to 40	**14**	**13**	**16**	**12**

The end of the data in the complete symbol is indicated by a Terminator consisting of between 3 and 9 zero bits (see Table 2), which is omitted or abbreviated if the remaining symbol capacity after the data bit stream is less than the required bit length of Terminator. The Terminator is not a mode indicator as such.

7.4.2 Extended Channel Interpretation (ECI) mode

7.4.2.1 General

This mode, used for encoding data subject to alternative interpretations of byte values (e.g. alternative character sets) in accordance with the AIM ECI specification which defines the pre-processing of this type of data, is invoked by the use of mode indicator **0111.**

The Extended Channel Interpretation can only be used with readers enabled to transmit the Symbology Identifier. Readers that cannot transmit the Symbology Identifier cannot transmit the data from any symbol containing an ECI.

Input ECI data shall be handled by the encoding system as a series of byte values.

Data in an ECI sequence may be encoded in whatever mode or modes permit the most efficient encoding of the byte values of the data, irrespective of their significance. For example, a sequence of bytes in the range 30_{HEX} to 39_{HEX} could be encoded in Numeric mode (see 7.4.3) as though it were a sequence of digits 0 – 9 even though it might not actually represent numeric data. In order to determine the value of the character count indicator, the number of bytes (or, in Kanji mode, of byte pairs) shall be used.

7.4.2.2 ECI Designator

Each Extended Channel Interpretation is designated by a six-digit assignment number which is encoded in the QR Code symbol as the first one, two or three codewords following the ECI mode indicator. The encoding rules are defined in Table 4. The ECI Designator appears in the data to be encoded as character $5C_{HEX}$ [\ or backslash (reverse solidus) in ISO/IEC 8859-1, ¥ or yen sign in JIS8] followed by the six digit assignment number. Where $5C_{HEX}$ appears as true data it shall be doubled in the data string before encoding in symbols to which the ECI protocol applies.

When a single occurrence of $5C_{HEX}$ is encountered in the input to the decoder, an ECI mode indicator is inserted followed by the ECI Designator. When a doubled $5C_{HEX}$ is encountered, it is encoded as two $5C_{HEX}$ bytes.

On decoding, the binary pattern of the first ECI Designator codeword (i.e. the codeword following the mode indicator in ECI mode), determines the length of the ECI Designator sequence. The number of **1** bits before the first **0** bit defines the number of additional codewords after the first used to represent the ECI Assignment number. The bit sequence after the first **0** bit is the binary representation of the ECI Assignment number. The lower numbered ECI assignments may be encoded in multiple ways, but the shortest way is preferred.

Table 4 — Encoding ECI Assignment Number

ECI Assignment Value	No. of Codewords	Codeword values
000000 to 000127	1	**0**bbbbbbb
000000 to 016383	2	**10**bbbbbb bbbbbbbb
000000 to 999999	3	**110**bbbbb bbbbbbbb bbbbbbbb
		where b … b is the binary value of the ECI Assignment number

Example

Assume data to be encoded is in Greek, using character set ISO/IEC 8859-7 (ECI 000009) in version 1-H symbol.

Data to be encoded: \000009**ΑΒΓΔΕ** (character values $A1_{HEX}$, $A2_{HEX}$, $A3_{HEX}$, $A4_{HEX}$, $A5_{HEX}$)

Bit sequence in symbol:

ECI mode indicator **0111**

ECI Assignment number (000009) **0 0001001**

Mode indicator (byte) **0100**

Character count indicator (5) **00000101**

Data: **10100001 10100010 10100011 10100100 10100101**

Final bit string: **0111 00001001 0100 00000101 10100001 10100010 10100011 10100100 10100101**

See 14.3 for example of transmission of this data following decoding.

7.4.2.3 Multiple ECIs

Refer to the AIM ECI specification for the rules defining the effect of a subsequent ECI Designator in an ECI data segment. For example, data to which a character set ECI has been applied may also be subject to encryption or compaction using a transformation ECI which will co-exist with the initial ECI, or a second character set ECI will have the effect of terminating the first ECI and starting a new ECI segment. Where any ECI Designator appears in the data, it shall be encoded in the QR Code symbol in accordance with 7.4.2.2 and shall commence a new mode segment.

7.4.2.4 ECIs and Structured Append

Any ECI(s) invoked shall apply subject to the rules defined above and in the AIM ECI specification until the end of the encoded data or a change of ECI (signalled by mode indicator **0111**). If the encoded data in the ECI(s) extends through two or more symbols in Structured Append mode, it is necessary to provide an ECI header consisting of ECI mode indicator and ECI Designator number for each ECI in force, immediately following the Structured Append header, in subsequent symbols in which the ECI continues in force.

7.4.3 Numeric mode

The input data string is divided into groups of three digits, and each group is converted to its 10-bit binary equivalent. If the number of input digits is not an exact multiple of three, the final one or two digits are converted to 4 or 7 bits respectively. The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the Numeric mode has either 4 bits for QR Code symbols or the number of bits defined in Table 2 for Micro QR Code symbols, and the character count indicator has the number of bits defined in Table 3. The number of input data characters is converted to its binary equivalent and added as the character count indicator after the mode indicator and before the binary data sequence.

EXAMPLE 1 (for Version 1-H symbol)

Input data: **01234567**

1. Divide into groups of three digits:	**012 345 67**
2. Convert each group to its binary equivalent:	**012 → 0000001100**
	345 → 0101011001
	67 → 1000011
3. Connect the binary data in sequence:	**0000001100 0101011001 1000011**

4. Convert character count indicator to binary (10 bits for version 1-H):

No. of input data characters: **8 → 0000001000**

5. Add mode indicator **0001** and character count indicator to binary data:

0001 0000001000 0000001100 0101011001 1000011

EXAMPLE 2 (for Micro QR Code version M3-M symbol)

Input data:	**0123456789012345**
1. Divide into groups of three digits:	**012 345 678 901 234 5**
2. Convert each group to its binary equivalent:	**012 = 0000001100**
	345 = 0101011001

678 = 1010100110

901 = 1110000101

234 = 0011101010

5 = 0101

3. Connect the binary data in sequence:

0000001100 0101011001 1010100110 1110000101 0011101010 0101

4. Convert character count indicator to binary (5 bits for version M3-M):

No. of input data characters: **16 = 10000**

5. Add mode indicator (**00** for version M3-M) and character count indicator to binary data:

00 10000 0000001100 0101011001 1010100110 1110000101 0011101010 0101

For any number of data characters the length of the bit stream in Numeric mode is given by the following formula:

$$B = M + C + 10(D\,\mathrm{DIV}\,3) + R$$

where:

B number of bits in bit stream

M number of bits in mode indicator (4 for QR Code symbols, or as shown in Table 2 for Micro QR Code symbols)

C number of bits in character count indicator (from Table 3)

D number of input data characters

R 0 if (D MOD 3) = 0

R 4 if (D MOD 3) = 1

R 7 if (D MOD 3) = 2

7.4.4 Alphanumeric mode

Each input data character is assigned a character value V from 0 to 44 according to Table 5.

Table 5 — Encoding/decoding table for Alphanumeric mode

Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value
0	0	6	6	C	12	I	18	O	24	U	30	SP	36	.	42
1	1	7	7	D	13	J	19	P	25	V	31	$	37	/	43
2	2	8	8	E	14	K	20	Q	26	W	32	%	38	:	44
3	3	9	9	F	15	L	21	R	27	X	33	*	39		
4	4	A	10	G	16	M	22	S	28	Y	34	+	40		
5	5	B	11	H	17	N	23	T	29	Z	35	-	41		

Input data characters are divided into groups of two characters which are encoded as 11-bit binary codes. The character value of the first character is multiplied by 45 and the character value of the second digit is added to the product. The sum is then converted to an 11-bit binary number. If the number of input data characters is not a multiple of two, the character value of the final character is encoded as a

6-bit binary number. The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the Alphanumeric mode has either 4 bits for QR Code symbols or the number of bits defined in Table 2 for Micro QR Code symbols, and the character count indicator has the number of bits defined in Table 3. The number of input data characters is converted to its binary equivalent and added as the character count indicator after the mode indicator and before the binary data sequence.

In FNC1 mode symbols the **FNC1** character may occur in the data. It is represented in Alphanumeric mode by the character %. Refer to 7.4.8.2, 7.4.8.3 and 14.4 for details of the encoding and transmission of **FNC1** and %.

EXAMPLE (for Version 1-H symbol)

Input data:	**AC-42**
1. Determine character values according to Table 5.	**AC-42 → (10,12,41,4,2)**
2. Divide the result into groups of two decimal values:	**(10,12) (41,4) (2)**
3. Convert each group to its 11-bit binary equivalent:	**(10,12) 10*45+12 → 462 → 00111001110**
	(41,4) 41*45+4 → 1849 → 11100111001
	(2) → 2 → 000010
4. Connect the binary data in sequence:	**00111001110 11100111001 000010**

5. Convert character count indicator to binary (9 bits for version 1-H):

No. of input data characters: **5 → 000000101**

6. Add mode indicator **0010** and character count indicator to binary data:

0010 000000101 00111001110 11100111001 000010

For any number of data characters the length of the bit stream in Alphanumeric mode is given by the following formula:

$$B = M + C + 11(D\,\mathrm{DIV}\,2) + 6(D\,\mathrm{MOD}\,2)$$

where:

B number of bits in bit stream

M number of bits in mode indicator (4 for QR Code symbols, or as shown in Table 2 for Micro QR Code symbols)

C number of bits in character count indicator (from Table 3)

D number of input data characters

7.4.5 Byte mode

In this mode, one 8-bit codeword directly represents the byte value of the input data character, i.e. a density of 8 bits/character.

Table 6 — Encoding/decoding table for ISO/IEC 8859-1 character set

Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.
0	NUL	32	space	64	@	96	`	128		160	NBSP	192	À	224	à

Table 6 *(continued)*

Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.	Byte	Char.
1	SOH	33	!	65	A	97	a	129		161	¡	193	Á	225	á
2	STX	34	"	66	B	98	b	130		162	¢	194	Â	226	â
3	ETX	35	#	67	C	99	c	131		163	£	195	Ã	227	ã
4	EOT	36	$	68	D	100	d	132		164	¤	196	Ä	228	ä
5	ENQ	37	%	69	E	101	e	133		165	¥	197	Å	229	å
6	ACK	38	&	70	F	102	f	134		166	¦	198	Æ	230	æ
7	BEL	39	'	71	G	103	g	135		167	§	199	Ç	231	ç
8	BS	40	(	72	H	104	h	136		168	¨	200	È	232	è
9	HT	41	)	73	I	105	I	137		169	©	201	É	233	é
10	LF	42	*	74	J	106	j	138		170	ª	202	Ê	234	ê
11	VT	43	+	75	K	107	k	139		171	«	203	Ë	235	ë
12	FF	44	,	76	L	108	l	140		172	¬	204	Ì	236	ì
13	CR	45	-	77	M	109	m	141		173	SHY	205	Í	237	í
14	SO	46	.	78	N	110	n	142		174	®	206	Î	238	î
15	SI	47	/	79	O	111	o	143		175	¯	207	Ï	239	ï
16	DLE	48	0	80	P	112	p	144		176	°	208	Ð	240	ð
17	DC1	49	1	81	Q	113	q	145		177	±	209	Ñ	241	ñ
18	DC2	50	2	82	R	114	r	146		178	²	210	Ò	242	ò
19	DC3	51	3	83	S	115	s	147		179	³	211	Ó	243	ó
20	DC4	52	4	84	T	116	t	148		180	´	212	Ô	244	ô
21	NAK	53	5	85	U	117	u	149		181	µ	213	Õ	245	õ
22	SYN	54	6	86	V	118	v	150		182	¶	214	Ö	246	ö
23	ETB	55	7	87	W	119	w	151		183	·	215	×	247	÷
24	CAN	56	8	88	X	120	x	152		184	¸	216	Ø	248	ø
25	EM	57	9	89	Y	121	y	153		185	¹	217	Ù	249	ù
26	SUB	58	:	90	Z	122	z	154		186	º	218	Ú	250	ú
27	ESC	59	;	91	[	123	{	155		187	»	219	Û	251	û
28	FS	60	<	92	\	124	|	156		188	¼	220	Ü	252	ü
29	GS	61	=	93	]	125	}	157		189	½	221	Ý	253	ý
30	RS	62	>	94	^	126	~	158		190	¾	222	Þ	254	þ
31	US	63	?	95	_	127	DEL	159		191	¿	223	ß	255	ÿ

NOTE 1 In the JIS8 character set (see Table H.1), byte values 80_{HEX} to $9F_{HEX}$ and $E0_{HEX}$ to FF_{HEX} are not assigned but are reserved values. Some of those values are used as the first byte in the Shift JIS character set (see Table H.2) and may be used to distinguish between the JIS8 and Shift JIS character sets, or to enable Kanji mode compaction to be carried out. JIS X 0208 gives details of the shift coded representation.

NOTE 2 Byte values 00_{HEX} to $7F_{HEX}$ in the JIS8 character set correspond to ISO/IEC 8859-1 and ISO/IEC 646 IRV, except values $5C_{HEX}$ and $7E_{HEX}$.

The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the Byte mode has either 4 bits for QR Code symbols or the number of bits defined in Table 2 for Micro QR Code symbols, and the character count indicator has the number of bits defined in Table 3. The number of input data characters is converted to its binary equivalent and added after the mode indicator and before the binary data sequence.

For any number of data characters the length of the bit stream in Byte mode is given by the following formula:

$$B = M + C + 8D$$

where:

B number of bits in bit stream

M number of bits in mode indicator (4 for QR Code symbols, or as shown in Table 2 for Micro QR Code symbols)

C number of bits in character count indicator (from Table 3)

D number of input data characters

7.4.6 Kanji mode

In the Shift JIS system, Kanji characters are represented by a two byte combination. These byte values are shifted from the JIS X 0208 values. JIS X 0208 gives details of the shift coded representation. Input data characters in Kanji mode are compacted to 13-bit binary codewords as defined below. The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the Numeric mode has either 4 bits for QR Code symbols or the number of bits defined in Table 2 for Micro QR Code symbols, and the character count indicator has the number of bits defined in Table 3. The number of input data characters is converted to its binary equivalent and added as the character count indicator after the mode indicator and before the binary data sequence.

1. For characters with Shift JIS values from 8140_{HEX} to $9FFC_{HEX}$:
   a) Subtract 8140_{HEX} from Shift JIS value;
   b) Multiply most significant byte of result by $C0_{HEX}$;
   c) Add least significant byte to product from b);
   d) Convert result to a 13-bit binary string.
2. For characters with Shift JIS values from $E040_{HEX}$ to $EBBF_{HEX}$:
   a) Subtract $C140_{HEX}$ from Shift JIS value;
   b) Multiply most significant byte of result by $C0_{HEX}$;
   c) Add least significant byte to product from b);
   d) Convert result to a 13-bit binary string.

EXAMPLES:

Input character	“点”	“茗”
(Shift JIS value):	935F	E4AA
1. Subtract 8140 or C140	935F - 8140 = 121F	E4AA - C140 = 236A
2. Multiply m.s.b. by C0	12 × C0 = D80	23 × C0 = 1A40
3. Add l.s.b.	D80 + 1F = D9F	1A40 + 6A = 1AAA
4. Convert to 13-bit binary	0D9F →0 1101 1001 1111	1AAA →1 1010 1010 1010

3. For all characters:

   e) Prefix binary sequence representing input data characters with mode indicator (from Table 2) and character count indicator binary equivalent (number of bits defined in Tables);

For any number of data characters the length of the bit stream in Kanji mode is given by the following formula:

$$B = M + C + 13D$$

where:

B number of bits in bit stream

M number of bits in mode indicator (4 for QR Code symbols, or as shown in Table 2 for Micro QR Code symbols)

C number of bits in character count indicator (from Table 3)

D number of input data characters

7.4.7 Mixing modes

There is the option for a symbol to contain sequences of data in one mode and then to change modes if the data content requires it, or in order to increase the density of encoding. Refer to Annex J for guidance. Each segment of data is encoded in the appropriate mode as indicated in 7.4.2 to 7.4.6, with the basic structure mode indicator/character count indicator/Data and followed immediately by the mode indicator commencing the next segment. Figure 13 illustrates the structure of data containing n segments.

Segment 1			Segment 2			……	Segment *n*			Terminator
mode indicator 1	character count indicator	Data	mode indicator 2	character count indicator	Data	……	mode indicator *n*	character count indicator	Data	

Figure 13 — Format of mixed mode data

7.4.8 FNC1 modes

7.4.8.1 General

In QR Code symbols, there are two mode indicators which are used cumulatively with those defined in 7.3.2 to 7.3.9 and 7.4.2 to 7.4.7 to identify symbols encoding messages formatted according to specific predefined industry or application specifications. These (together with any associated parameter data) precede the mode indicator(s) used to encode the data efficiently. When these mode indicators are used, it is necessary for the decoder to transmit the Symbology Identifier as defined in 14.2 and Annex F.

7.4.8.2 FNC1 in first position

NOTE "first position" is not used in a literal sense but is a historical reference to the position of the FNC1 symbol character in Code 128 symbols.

This mode indicator identifies symbols encoding data formatted according to the GS1 Application Identifiers standard. For this purpose, it shall only be used once in a symbol and shall be placed immediately before the first mode indicator used for efficient data encoding (Numeric, Alphanumeric, Byte or Kanji), and after any ECI or Structured Append header. Where the GS1 specifications call for the FNC1 character (in other symbologies which use this special character) to be used as a data field separator (i.e. at the end of a variable-length data field), QR Code symbols shall use the % character in

Alphanumeric mode or character **GS** (byte value $1D_{HEX}$) in Byte mode to perform this function. If the % character occurs as part of the data it shall be encoded as %%. Decoders encountering % in these symbols shall transmit it as ASCII/JIS8 value $1D_{HEX}$, and if %% is encountered it shall be transmitted as a single % character.

EXAMPLE 1

Input data: **0104912345123459** (Application Identifier 01 = GS1 article no., fixed length; data: 04912345123459)

15970331 (Application Identifier 15 = "Best before" date YYMMDD, fixed length; data: 31 March 1997)

30128 (Application Identifier 30 = quantity, variable length; data: 128) (requires separator character)

10ABC123 (Application Identifier 10 = batch number, variable length; data: ABC123)

Data to be encoded:

01049123451234591597033130128%10ABC123

Bit sequence in symbol:

0101 (mode indicator, FNC1 implied in 1st position)

0001 (mode indicator, Numeric mode)

0000011101 (character count indicator, 29)

<data bits for **01049123451234591597033130128**>

0010 (mode indicator, Alphanumeric mode)

000001001 (character count indicator, 9)

<data bits for **%10ABC123**>

Transmitted data (see 14.2 and Annex F)

]Q301049123451234591597033130128**<1D$_{HEX}$>**10ABC123

EXAMPLE 2 Encoding/transmission of % character in data:

Input data: **123%**

Encoded as: **123%%**

Transmitted as: **123%**

7.4.8.3 FNC1 in second position

NOTE "second position" is not used in a literal sense but is a historical reference to the position of the FNC1 symbol character in Code 128 symbols.

This mode indicator identifies symbols formatted in accordance with specific industry or application specifications previously agreed with AIM International. It is immediately followed by a one-byte codeword the value of which is that of the Application Indicator assigned to identify the specification concerned by AIM International. For this purpose, it shall only be used once in a symbol and shall be placed immediately before the first mode indicator used for efficient data encoding (Numeric, Alphanumeric, Byte or Kanji), and after any ECI or structured Append header. An Application Indicator may take the form of any single Latin alphabetic character from the set {a - z, A - Z} (represented by the

ASCII value of the character plus 100) or a two-digit number (represented by its numeric value directly) and shall be transmitted by the decoder as the first one or two characters immediately preceding the data. Where the application specifications call for the FNC1 character (in other symbologies which use this special character) to be used as a data field separator, QR Code symbols shall use the % character in Alphanumeric mode or character **GS** (ASCII/JIS8 value $1D_{HEX}$) in Byte mode to perform this function. If the % character occurs as part of the data it shall be encoded as %%. Decoders encountering % in these symbols shall transmit it as ASCII/JIS8 value $1D_{HEX}$, and if %% is encountered it shall be transmitted as a single % character.

EXAMPLE:

NOTE Application Indicator 37 has not been assigned at the time of publication to any organisation and the data content of the example is purely arbitrary.

Application Indicator: 37

Input data: AA1234BBB112text text text text<CR>

Bit sequence in symbol:

1001 (mode indicator, FNC1 implied in 2nd position)

00100101 (Application Indicator, 37)

0010 (mode indicator, Alphanumeric mode)

000001100 (character count indicator, 12)

<data bits for **AA1234BBB112**>

0100 (mode indicator, Byte mode)

00010100 (character count indicator, 20)

<data bits for **text text text text<CR>** >

Transmitted data:

]Q537AA1234BBB112text text text text<CR>

7.4.9 Terminator

The end of data in the symbol is signalled by the Terminator sequence of **0** bits, as defined in Table 2, appended to the data bit stream following the final mode segment. The Terminator shall be omitted if the data bit stream completely fills the capacity of the symbol, or abbreviated if the remaining capacity of the symbol is less than the required bit length of Terminator.

7.4.10 Bit stream to codeword conversion

The bit streams corresponding to each mode segment shall be connected in order. The Terminator shall be appended to the complete bit stream as defined in 7.4.9. The resulting message bit stream shall then be divided into codewords. All codewords are 8 bits in length, except for the final data symbol character in Micro QR Code versions M1 and M3 symbols, which is 4 bits in length. If the bit stream length is such that it does not end at a codeword boundary, padding bits with binary value 0 shall be added after the final bit (least significant bit) of the data stream to extend it to the codeword boundary. The message bit stream shall then be extended to fill the data capacity of the symbol corresponding to the Version and Error Correction Level, as defined in Table 8, by adding the Pad Codewords **11101100** and **00010001** alternately. For Micro QR Code versions M1 and M3 symbols, the final data codeword is 4 bits long. The Pad Codeword used in the final data symbol character position in Micro QR Code versions M1 and M3 symbols shall be represented as **0000**. The resulting series of codewords, the data codeword sequence, is then processed as described in 7.5 to add error correction codewords to the message. In certain versions

of symbol, it may be necessary to add 3, 4 or 7 Remainder Bits (all zeros) to the end of the message, after the final error correction codeword, in order exactly to fill the symbol capacity (see Table 1).

Table 7 — Number of symbol characters and input data capacity for QR Code

Version	Error correction level	Number of data codewords	Number of data bits	Data capacity			
				Numeric	Alphanumeric	Byte	Kanji
M1	Error Detection only	3	20	5	-	-	-
M2	L M	5 4	40 32	10 8	6 5	- -	- -
M3	L M	11 9	84 68	23 18	14 11	9 7	6 4
M4	L M Q	16 14 10	128 112 80	35 30 21	21 18 13	15 13 9	9 8 5
1	L M Q H	19 16 13 9	152 128 104 72	41 34 27 17	25 20 16 10	17 14 11 7	10 8 7 4
2	L M Q H	34 28 22 16	272 224 176 128	77 63 48 34	47 38 29 20	32 26 20 14	20 16 12 8
3	L M Q H	55 44 34 26	440 352 272 208	127 101 77 58	77 61 47 35	53 42 32 24	32 26 20 15
4	L M Q H	80 64 48 36	640 512 384 288	187 149 111 82	114 90 67 50	78 62 46 34	48 38 28 21
5	L M Q H	108 86 62 46	864 688 496 368	255 202 144 106	154 122 87 64	106 84 60 44	65 52 37 27
6	L M Q H	136 108 76 60	1 088 864 608 480	322 255 178 139	195 154 108 84	134 106 74 58	82 65 45 36
7	L M Q H	156 124 88 66	1 248 992 704 528	370 293 207 154	224 178 125 93	154 122 86 64	95 75 53 39
8	L M Q H	194 154 110 86	1 552 1 232 880 688	461 365 259 202	279 221 157 122	192 152 108 84	118 93 66 52
9	L M Q H	232 182 132 100	1 856 1 456 1 056 800	552 432 312 235	335 262 189 143	230 180 130 98	141 111 80 60

Table 7 *(continued)*

Version	Error correction level	Number of data codewords	Number of data bits	Data capacity			
				Numeric	Alphanumeric	Byte	Kanji
10	L	274	2 192	652	395	271	167
	M	216	1 728	513	311	213	131
	Q	154	1 232	364	221	151	93
	H	122	976	288	174	119	74
11	L	324	2 592	772	468	321	198
	M	254	2 032	604	366	251	155
	Q	180	1 440	427	259	177	109
	H	140	1 120	331	200	137	85
12	L	370	2 960	883	535	367	226
	M	290	2 320	691	419	287	177
	Q	206	1 648	489	296	203	125
	H	158	1 264	374	227	155	96
13	L	428	3 424	1 022	619	425	262
	M	334	2 672	796	483	331	204
	Q	244	1 952	580	352	241	149
	H	180	1 440	427	259	177	109
14	L	461	3 688	1 101	667	458	282
	M	365	2 920	871	528	362	223
	Q	261	2 088	621	376	258	159
	H	197	1 576	468	283	194	120
15	L	523	4 184	1 250	758	520	320
	M	415	3 320	991	600	412	254
	Q	295	2 360	703	426	292	180
	H	223	1 784	530	321	220	136
16	L	589	4 712	1 408	854	586	361
	M	453	3 624	1 082	656	450	277
	Q	325	2 600	775	470	322	198
	H	253	2 024	602	365	250	154
17	L	647	5 176	1 548	938	644	397
	M	507	4 056	1 212	734	504	310
	Q	367	2 936	876	531	364	224
	H	283	2 264	674	408	280	173
18	L	721	5 768	1 725	1 046	718	442
	M	563	4 504	1 346	816	560	345
	Q	397	3 176	948	574	394	243
	H	313	2 504	746	452	310	191
19	L	795	6 360	1 903	1 153	792	488
	M	627	5 016	1 500	909	624	384
	Q	445	3 560	1 063	644	442	272
	H	341	2 728	813	493	338	208
20	L	861	6 888	2 061	1 249	858	528
	M	669	5 352	1 600	970	666	410
	Q	485	3 880	1 159	702	482	297
	H	385	3 080	919	557	382	235
21	L	932	7 456	2 232	1 352	929	572
	M	714	5 712	1 708	1 035	711	438
	Q	512	4 096	1 224	742	509	314
	H	406	3 248	969	587	403	248
22	L	1 006	8 048	2 409	1 460	1 003	618
	M	782	6 256	1 872	1 134	779	480
	Q	568	4 544	1 358	823	565	348
	H	442	3 536	1 056	640	439	270

Table 7 *(continued)*

Version	Error correction level	Number of data codewords	Number of data bits	Data capacity			
				Numeric	Alphanumeric	Byte	Kanji
23	L M Q H	1 094 860 614 464	8 752 6 880 4 912 3 712	2 620 2 059 1 468 1 108	1 588 1 248 890 672	1 091 857 611 461	672 528 376 284
24	L M Q H	1 174 914 664 514	9 392 7 312 5 312 4 112	2 812 2 188 1 588 1 228	1 704 1 326 963 744	1 171 911 661 511	721 561 407 315
25	L M Q H	1 276 1 000 718 538	10 208 8 000 5 744 4 304	3 057 2 395 1 718 1 286	1 853 1 451 1 041 779	1 273 997 715 535	784 614 440 330
26	L M Q H	1 370 1 062 754 596	10 960 8 496 6 032 4 768	3 283 2 544 1 804 1 425	1 990 1 542 1 094 864	1 367 1 059 751 593	842 652 462 365
27	L M Q H	1 468 1 128 808 628	11 744 9 024 6 464 5 024	3 517 2 701 1 933 1 501	2 132 1 637 1 172 910	1 465 1 125 805 625	902 692 496 385
28	L M Q H	1 531 1 193 871 661	12 248 9 544 6 968 5 288	3 669 2 857 2 085 1 581	2 223 1 732 1 263 958	1 528 1 190 868 658	940 732 534 405
29	L M Q H	1 631 1 267 911 701	13 048 10 136 7 288 5 608	3 909 3 035 2 181 1 677	2 369 1 839 1 322 1 016	1 628 1 264 908 698	1 002 778 559 430
30	L M Q H	1 735 1 373 985 745	13 880 10 984 7 880 5 960	4 158 3 289 2 358 1 782	2 520 1 994 1 429 1 080	1 732 1 370 982 742	1 066 843 604 457
31	L M Q H	1 843 1 455 1 033 793	14 744 11 640 8 264 6 344	4 417 3 486 2 473 1 897	2 677 2 113 1 499 1 150	1 840 1 452 1 030 790	1 132 894 634 486
32	L M Q H	1 955 1 541 1 115 845	15 640 12 328 8 920 6 760	4 686 3 693 2 670 2 022	2 840 2 238 1 618 1 226	1 952 1 538 1 112 842	1 201 947 684 518
33	L M Q H	2 071 1 631 1 171 901	16 568 13 048 9 368 7 208	4 965 3 909 2 805 2 157	3 009 2 369 1 700 1 307	2 068 1 628 1 168 898	1 273 1 002 719 553
34	L M Q H	2 191 1 725 1 231 961	17 528 13 800 9 848 7 688	5 253 4 134 2 949 2 301	3 183 2 506 1 787 1 394	2 188 1 722 1 228 958	1 347 1 060 756 590
35	L M Q H	2 306 1 812 1 286 986	18 448 14 496 10 288 7 888	5 529 4 343 3 081 2 361	3 351 2 632 1 867 1 431	2 303 1 809 1 283 983	1 417 1 113 790 605

Table 7 *(continued)*

Version	Error correction level	Number of data codewords	Number of data bits	Data capacity			
				Numeric	Alphanumeric	Byte	Kanji
36	L	2 434	19 472	5 836	3 537	2 431	1 496
	M	1 914	15 312	4 588	2 780	1 911	1 176
	Q	1 354	10 832	3 244	1 966	1 351	832
	H	1 054	8 432	2 524	1 530	1 051	647
37	L	2 566	20 528	6 153	3 729	2 563	1 577
	M	1 992	15 936	4 775	2 894	1 989	1 224
	Q	1 426	11 408	3 417	2 071	1 423	876
	H	1 096	8 768	2 625	1 591	1 093	673
38	L	2 702	21 616	6 479	3 927	2 699	1 661
	M	2 102	16 816	5 039	3 054	2 099	1 292
	Q	1 502	12 016	3 599	2 181	1 499	923
	H	1 142	9 136	2 735	1 658	1 139	701
39	L	2 812	22 496	6 743	4 087	2 809	1 729
	M	2 216	17 728	5 313	3 220	2 213	1 362
	Q	1 582	12 656	3 791	2 298	1 579	972
	H	1 222	9 776	2 927	1 774	1 219	750
40	L	2 956	23 648	7 089	4 296	2 953	1 817
	M	2 334	18 672	5 596	3 391	2 331	1 435
	Q	1 666	13 328	3 993	2 420	1 663	1 024
	H	1 276	10 208	3 057	1 852	1 273	784

NOTE 1 All codewords shall be 8 bits in length, except that the final data codeword for Versions M1 and M3 is 4 bits long.

NOTE 2 The number of Data Bits includes bits for mode indicator and character count indicator.

7.5 Error correction

7.5.1 Error correction capacity

QR Code employs Reed-Solomon error control coding to detect and correct errors. A series of error correction codewords is generated, which are added to the data codeword sequence in order to enable the symbol to withstand damage without loss of data. There are four user-selectable levels of error correction, as shown in Table 8, offering the capability of recovery from the following amounts of damage:

Table 8 — Error correction levels

Error Correction Level	Recovery Capacity % (approx.)
L	7
M	15
Q	25
H	30

Annex K.2 gives guidance on the appropriate level of error correction to be applied to a symbol.

Error correction level H is not available in Micro QR Code symbols.

The error correction codewords can correct two types of erroneous codewords, erasures (erroneous codewords at known locations) and errors (erroneous codewords at unknown locations). An erasure is an unscanned or undecodable symbol character. An error is a misdecoded symbol character. Since QR Code is a matrix symbology, a defect converting a module from dark to light or vice versa will result in

the affected symbol character misdecoding as an apparently valid but different codeword. Such an error causing a substitution error in the data requires two error correction codewords to correct it.

The number of erasures and errors correctable is given by the following formula:

$$e + 2t \leq d - p$$

where:

e number of erasures

t number of errors

d number of error correction codewords

p number of misdecode protection codewords

In the general case, $p = 0$. However, if most of the error correction capacity is used to correct erasures, then the possibility of an undetected error is increased. Whenever the number of erasures is more than half the number of error correction codewords, $p = 3$. For small symbols with less than 8 error correction codewords, erasure correction should not be used ($e = 0$ and $p > 0$).

For example, in a version 6-H symbol there is a total of 172 codewords, of which 112 are error correction codewords (leaving 60 data codewords). The 112 error correction codewords can correct 56 misdecodes or substitution errors, i.e. 56/172 or 32,6% of the symbol capacity.

In the formula above, the following values should be assigned to p:

— $p = 3$ in version 1-L and M2-L symbols,

— $p = 2$ in version 1-M, 2-L, M1, M2-M, M3-L, and M4-L symbols,

— $p = 1$ in version 1-Q, 1-H and 3-L symbols,

— $p = 0$ in all other cases.

Where $p > 0$ there are p (i.e. 1, 2 or 3) codewords which act as error detection codewords and prevent transmission of data from symbols where the number of errors exceeds the error correction capacity, e must be less than $d/2$. In a Version 2-L symbol, for example, the total number of codewords is 44; of these, 34 are data codewords and 10 error correction codewords. From Table 9 it can be seen that the error correction capacity is 4 errors (where $e = 0$). Substituting in the formula above,

$$0 + (2 \times 4) = 10 - 2$$

meaning that the correction of the 4 errors requires only 8 error correction codewords; the remaining 2 error correction codewords can therefore detect (but not correct) any additional errors and the symbol would, if there were more than 4 errors, fail to decode.

Depending on the Version and Error Correction Level, the data codeword sequence shall be subdivided into one or more blocks, to each of which the error correction algorithm shall be applied separately. Table 9 lists, for each version and Error Correction Level, the total number of codewords, the total number of error correction codewords, and the structure and number of error correction blocks.

If Remainder Bits are required to fill remaining modules in the symbol capacity for certain symbol versions they shall all be 0 bits.

Table 9 — Error correction characteristics for QR Code

Version	Total number of codewords	Error correction level	Number of error correction codewords	Value of p	Number of error correction blocks	Error correction code per block (c, k, r)[a]
M1	5	Error detection only	2	2	1	(5,3,0)[b]
M2	10	L M	5 6	3 2	1 1	(10,5,1)[b] (10,4,2)[b]
M3	17	L M	6 8	2	1 1	(17,11,2)[b] (17,9,4)
M4	24	L M Q	8 10 14	2 0 0	1 1 1	(24,16,3)[b] (24,14,5) (24,10,7)
1	26	L M Q H	7 10 13 17	3 2 1 1	1 1 1 1	(26,19,2)[b] (26,16,4)[b] (26,13,6)[b] (26,9,8)[b]
2	44	L M Q H	10 16 22 28	2 0 0 0	1 1 1 1	(44,34,4)[b] (44,28,8) (44,22,11) (44,16,14)
3	70	L M Q H	15 26 36 44	1 0 0 0	1 1 2 2	(70,55,7)[b] (70,44,13) (35,17,9) (35,13,11)
4	100	L M Q H	20 36 52 64	0	1 2 2 4	(100,80,10) (50,32,9) (50,24,13) (25,9,8)
5	134	L M Q H	26 48 72 88	0	1 2 2 2 2 2	(134,108,13) (67,43,12) (33,15,9) (34,16,9) (33,11,11) (34,12,11)
6	172	L M Q H	36 64 96 112	0	2 4 4 4	(86,68,9) (43,27,8) (43,19,12) (43,15,14)
7	196	L M Q H	40 72 108 130	0	2 4 2 4 4 1	(98,78,10) (49,31,9) (32,14,9) (33,15,9) (39,13,13) (40,14,13)
8	242	L M Q H	48 88 132 156	0	2 2 2 4 2 4 2	(121,97,12) (60,38,11) (61,39,11) (40,18,11) (41,19,11) (40,14,13) (41,15,13)

[a] c = total number of codewords, k = number of data codewords, r = error correction capacity

[b] Error correction capacity is less than half the number of error correction codewords to reduce the probability of misdecodes

Table 9 *(continued)*

Version	Total number of codewords	Error correction level	Number of error correction codewords	Value of p	Number of error correction blocks	Error correction code per block (c, k, r)[a]
9	292	L M Q H	60 110 160 192	0	2 3 2 4 4 4 4	(146,116,15) (58,36,11) (59,37,11) (36,16,10) (37,17,10) (36,12,12) (37,13,12)
10	346	L M Q H	72 130 192 224	0	2 2 4 1 6 2 6 2	(86,68,9) (87,69,9) (69,43,13) (70,44,13) (43,19,12) (44,20,12) (43,15,14) (44,16,14)
11	404	L M Q H	80 150 224 264	0	4 1 4 4 4 3 8	(101,81,10) (80,50,15) (81,51,15) (50,22,14) (51,23,14) (36,12,12) (37,13,12
12	466	L M Q H	96 176 260 308	0	2 2 6 2 4 6 7 4	(116,92,12) (117,93,12) (58,36,11) (59,37,11) (46,20,13) (47,21,13) (42,14,14) (43,15,14)
13	532	L M Q H	104 198 288 352	0	4 8 1 8 4 12 4	(133,107,13) (59,37,11) (60,38,11) (44,20,12) (45,21,12) (33,11,11) (34,12,11)
14	581	L M Q H	120 216 320 384	0	3 1 4 5 11 5 11 5	(145,115,15) (146,116,15) (64,40,12) (65,41,12) (36,16,10) (37,17,10) (36,12,12) (37,13,12)

[a] c = total number of codewords, k = number of data codewords, r = error correction capacity

[b] Error correction capacity is less than half the number of error correction codewords to reduce the probability of misdecodes

Table 9 *(continued)*

Version	Total number of codewords	Error correction level	Number of error correction codewords	Value of p	Number of error correction blocks	Error correction code per block (c, k, r)[a]
15	655	L M Q H	132 240 360 432	0	5 1 5 5 5 7 11 7	(109,87,11) (110,88,11) (65,41,12) (66,42,12) (54,24,15) (55,25,15) (36,12,12) (37,13,12)
16	733	L M Q H	144 280 408 480	0	5 1 7 3 15 2 3 13	(122,98,12) (123,99,12) (73,45,14) (74,46,14) (43,19,12) (44,20,12) (45,15,15) (46,16,15)
17	815	L M Q H	168 308 448 532	0	1 5 10 1 1 15 2 17	(135,107,14) (136,108,14) (74,46,14) (75,47,14) (50,22,14) (51,23,14) (42,14,14) (43,15,14)
18	901	L M Q H	180 338 504 588	0	5 1 9 4 17 1 2 19	(150,120,15) (151,121,15) (69,43,13) (70,44,13) (50,22,14) (51,23,14) (42,14,14) (43,15,14)
19	991	L M Q H	196 364 546 650	0	3 4 3 11 17 4 9 16	(141,113,14) (142,114,14) (70,44,13) (71,45,13) (47,21,13) (48,22,13) (39,13,13) (40,14,13)
20	1 085	L M Q H	224 416 600 700	0	3 5 3 13 15 5 15 10	(135,107,14) (136,108,14) (67,41,13) (68,42,13) (54,24,15) (55,25,15) (43,15,14) (44,16,14)

[a] c = total number of codewords, k = number of data codewords, r = error correction capacity

[b] Error correction capacity is less than half the number of error correction codewords to reduce the probability of misdecodes

Table 9 *(continued)*

Version	Total number of codewords	Error correction level	Number of error correction codewords	Value of p	Number of error correction blocks	Error correction code per block (c, k, r)[a]
21	1 156	L M Q H	224 442 644 750	0	4 4 17 17 6 19 6	(144,116,14) (145,117,14) (68,42,13) (50,22,14) (51,23,14) (46,16,15) (47,17,15)
22	1 258	L M Q H	252 476 690 816	0	2 7 17 7 16 34	(139,111,14) (140,112,14) (74,46,14) (54,24,15) (55,25,15) (37,13,12)
23	1 364	L M Q H	270 504 750 900	0	4 5 4 14 11 14 16 14	(151,121,15) (152,122,15) (75,47,14) (76,48,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
24	1 474	L M Q H	300 560 810 960	0	6 4 6 14 11 16 30 2	(147,117,15) (148,118,15) (73,45,14) (74,46,14) (54,24,15) (55,25,15) (46,16,15) (47,17,15)
25	1 588	L M Q H	312 588 870 1050	0	8 4 8 13 7 22 22 13	(132,106,13) (133,107,13) (75,47,14) (76,48,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
26	1 706	L M Q H	336 644 952 1110	0	10 2 19 4 28 6 33 4	(142,114,14) (143,115,14) (74,46,14) (75,47,14) (50,22,14) (51,23,14) (46,16,15) (47,17,15)

[a] c = total number of codewords, k = number of data codewords, r = error correction capacity

[b] Error correction capacity is less than half the number of error correction codewords to reduce the probability of misdecodes

Table 9 *(continued)*

Version	Total number of codewords	Error correction level	Number of error correction codewords	Value of p	Number of error correction blocks	Error correction code per block (c, k, r)[a]
27	1 828	L M Q H	360 700 1 020 1 200	0	8 4 22 3 8 26 12 28	(152,122,15) (153,123,15) (73,45,14) (74,46,14) (53,23,15) (54,24,15) (45,15,15) (46,16,15)
28	1 921	L M Q H	390 728 1 050 1 260	0	3 10 3 23 4 31 11 31	(147,117,15) (148,118,15) (73,45,14) (74,46,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
29	2 051	L M Q H	420 784 1 140 1 350	0	7 7 21 7 1 37 19 26	(146,116,15) (147,117,15) (73,45,14) (74,46,14) (53,23,15) (54,24,15) (45,15,15) (46,16,15)
30	2 185	L M Q H	450 812 1 200 1 440	0	5 10 19 10 15 25 23 25	(145,115,15) (146,116,15) (75,47,14) (76,48,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
31	2 323	L M Q H	480 868 1 290 1 530	0	13 3 2 29 42 1 23 28	(145,115,15) (146,116,15) (74,46,14) (75,47,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
32	2 465	L M Q H	510 924 1 350 1 620	0	17 10 23 10 35 19 35	(145,115,15) (74,46,14) (75,47,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)

[a] c = total number of codewords, k = number of data codewords, r = error correction capacity

[b] Error correction capacity is less than half the number of error correction codewords to reduce the probability of misdecodes

Table 9 *(continued)*

Version	Total number of codewords	Error correction level	Number of error correction codewords	Value of p	Number of error correction blocks	Error correction code per block (c, k, r)[a]
33	2 611	L M Q H	540 980 1 440 1 710	0	17 1 14 21 29 19 11 46	(145,115,15) (146,116,15) (74,46,14) (75,47,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
34	2 761	L M Q H	570 1 036 1 530 1 800	0	13 6 14 23 44 7 59 1	(145,115,15) (146,116,15) (74,46,14) (75,47,14) (54,24,15) (55,25,15) (46,16,15) (47,17,15)
35	2 876	L M Q H	570 1 064 1 590 1 890	0	12 7 12 26 39 14 22 41	(151,121,15) (152,122,15) (75,47,14) (76,48,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
36	3 034	L M Q H	600 1 120 1 680 1 980	0	6 14 6 34 46 10 2 64	(151,121,15) (152,122,15) (75,47,14) (76,48,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
37	3 196	L M Q H	630 1 204 1 770 2 100	0	17 4 29 14 49 10 24 46	(152,122,15) (153,123,15) (74,46,14) (75,47,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
38	3 362	L M Q H	660 1 260 1 860 2 220	0	4 18 13 32 48 14 42 32	(152,122,15) (153,123,15) (74,46,14) (75,47,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)

[a] c = total number of codewords, k = number of data codewords, r = error correction capacity

[b] Error correction capacity is less than half the number of error correction codewords to reduce the probability of misdecodes

Table 9 *(continued)*

Version	Total number of codewords	Error correction level	Number of error correction codewords	Value of p	Number of error correction blocks	Error correction code per block (c, k, r)[a]
39	3 532	L M Q H	720 1 316 1 950 2 310	0	20 4 40 7 43 22 10 67	(147,117,15) (148,118,15) (75,47,14) (76,48,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)
40	3 706	L M Q H	750 1 372 2 040 2 430	0	19 6 18 31 34 34 20 61	(148,118,15) (149,119,15) (75,47,14) (76,48,14) (54,24,15) (55,25,15) (45,15,15) (46,16,15)

[a] c = total number of codewords, k = number of data codewords, r = error correction capacity

[b] Error correction capacity is less than half the number of error correction codewords to reduce the probability of misdecodes

7.5.2 Generating the error correction codewords

The data codewords including Pad codewords as necessary shall be divided into the number of blocks shown in Table 9. Error correction codewords shall be calculated for each block and appended to the data codewords.

NOTE Micro QR Code symbols consist of a single block.

The polynomial arithmetic for QR Code shall be calculated using bit-wise modulo 2 arithmetic and byte-wise modulo 100011101 arithmetic. This is a Galois field of 2^8 with 100011101 representing the field's prime modulus polynomial $x^8 + x^4 + x^3 + x^2 + 1$.

The data codewords are the coefficients of the terms of a polynomial with the coefficient of the highest term being the first data codeword and that of the lowest power term being the last data codeword before the first error correction codeword.

The error correction codewords are the remainder after dividing the data codewords by a polynomial *g(x)* used for error correction codes (see Annex A). The highest order coefficient of the remainder is the first error correction codeword and the zero power coefficient is the last error correction codeword and the last codeword in the block.

NOTE If this calculation is performed by "long division" the symbol data polynomial must first be multiplied by x^k.

Thirty-six different generator polynomials are used for generating the error correction codewords for QR Code. These are given in Annex A.

This can be implemented by using the division circuit as shown in Figure 14. The registers b_0 through b_{k-1} are initialized as zeros. There are two phases to generate the encoding. In the first phase, with the switch in the down position the data codewords are passed both to the output and the circuit. The first phase is complete after *n* clock pulses. In the second phase ($n + 1 \ldots n + k$ clock pulses), with the switch in the up position, the error correction codewords $\varepsilon_{k-1} \ldots \varepsilon_0$ are generated by flushing the registers in order while keeping the data input at 0.

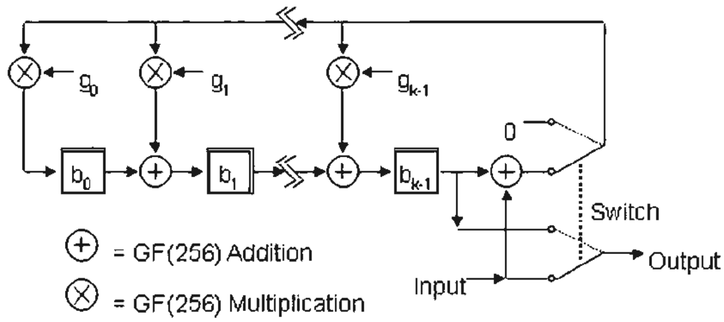


Figure 14 — Error correction codeword encoding circuit

7.6 Constructing the final message codeword sequence

The total number of codewords in the message shall always be equal to the total number of codewords capable of being represented in the symbol, as shown in Tables 7 and 9.

The following steps shall be followed to construct the final sequence of codewords (data plus error correction codewords plus Remainder Codewords if necessary):

1. Divide the data codeword sequence into *n* blocks as defined in Table 9 according to the version and error correction level (or a single block for Micro QR Code symbols).

2. For each data block, calculate a corresponding block of error correction codewords as defined in 7.5.2 and Annex A.

3. Assemble the final sequence by taking data and error correction codewords from each block in turn. For example, if there are four blocks the sequence would be: data block 1, codeword 1; data block 2, codeword 1; ... ; data block 4, codeword 1; data block 1, codeword 2; ... and similarly to data block 3, final codeword; data block 4, final codeword; then error correction block 1, codeword 1, error correction block 2, codeword 1, ... and similarly to error correction block 4, final codeword. QR Code symbols contain data and error correction blocks which always exactly fill the symbol codeword capacity. In certain QR Code versions, however, where the number of modules available for data and error correction codewords is not an exact multiple of 8, there may be a need for 3, 4 or 7 Remainder Bits to be appended to the final message bit stream in order to fill exactly the number of modules in the encoding region.

The shortest data block (or blocks) shall be placed first in the sequence and all the data codewords shall be placed in the symbol before the first error correction codeword. For example, the Version 5-H symbol comprises four data and four error correction blocks, the first two of each of which contain 11 data and 22 error correction codewords respectively, while the third and fourth pairs of blocks contain 12 data and 22 error correction codewords respectively. In this symbol, the character arrangement can be depicted as shown in Figure 15. Each row of the figure corresponds to one block of data codewords (shown as D_n) followed by the associated block of error correction codewords (shown as E_n); the sequence of character placement in the symbol is obtained by reading down each column of the figure in turn.

	Data codewords					Error correction codewords			
Block 1	D_1	D_2	.....	D_{11}		E_1	E_2	.....	E_{22}
Block 2	D_{12}	D_{13}	.....	D_{22}		E_{23}	E_{24}	.....	E_{44}
Block 3	D_{23}	D_{24}	.....	D_{33}	D_{34}	E_{45}	E_{46}	.....	E_{66}
Block 4	D_{35}	D_{36}	.....	D_{45}	D_{46}	E_{67}	E_{68}	.....	E_{88}

Figure 15 — Constructing the final message codeword sequence

The final message codeword sequence for the Version 5-H symbol is therefore:

D_1, D_{12}, D_{23}, D_{35}, D_2, D_{13}, D_{24}, D_{36}, ... D_{11}, D_{22}, D_{33}, D_{45}, D_{34}, D_{46}, E_1, E_{23}, E_{45}, E_{67}, E_2, E_{24}, E_{46}, E_{68}, ... E_{22}, E_{44}, E_{66}, E_{88}. The symbol module capacity is filled by adding 7 Remainder (0) bits as needed after the final codeword.

7.7 Codeword placement in matrix

7.7.1 Symbol character representation

There are two types of symbol character, regular and irregular, in the QR Code symbol. Their use depends on their position in the symbol, relative to other symbol characters and function patterns.

Most codewords shall be represented in a regular 2 × 4 module block in the symbol. There are two ways of positioning these blocks, in a vertical arrangement (2 modules wide and 4 modules high) and, if necessary when placement changes direction, in a horizontal arrangement (4 modules wide and 2 modules high). Irregular symbol characters are used when changing direction or in the vicinity of alignment or other function Patterns. Examples are shown in Figures 16, 17 and 18.

7.7.2 Function pattern placement

A square blank matrix shall be constructed with the number of modules horizontally and vertically corresponding to the Version in use. Positions corresponding to the finder pattern, separator, timing pattern, and alignment patterns shall be filled with either dark modules or light modules as appropriate. Module positions for the format information and version information shall be left temporarily blank. These blank positions are shown in Figures 19 and 20 and are common to all Versions (although the version information is not present in Version 1 to 6 symbols). Annex E defines the positioning of alignment patterns.

7.7.3 Symbol character placement

In the encoding region of the QR Code symbol, symbol characters are positioned in two-module wide columns commencing at the lower right corner of the symbol and running alternately upwards and downwards from the right to the left. The principles governing the placement of characters and of bits within the characters are given below. Figures 19 and 20 illustrate Version 2 and Version 7 symbols applying these principles.

a) The sequence of bit placement in the column shall be from right to left and either upwards or downwards in accordance with the direction of symbol character placement.

b) The most significant bit (shown as bit 7) of each codeword shall be placed in the first available module position. Subsequent bits shall be placed in the next module positions. The most significant bit therefore occupies the lower right module of a regular symbol character when the direction of

placement is upwards, and the upper right module when the direction of placement is downwards. It may however occupy the lower left module of an irregular symbol character if the previous character has ended in the right-hand module column (see Figure 18).

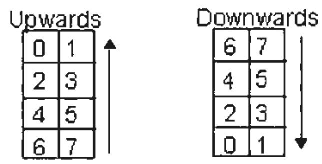


Figure 16 — Bit placement in regular symbol character in upwards and downwards directions

c) When a symbol character encounters the horizontal boundary of an alignment pattern or of the timing pattern in both module columns, it shall continue above or below the pattern as though the encoding region were continuous.

d) When the upper or lower boundary of the symbol character region is reached (i.e. the edge of the symbol, format information, version information, or separator) any remaining bits in the codeword shall be placed in the next column to the left. The direction of placement reverses.

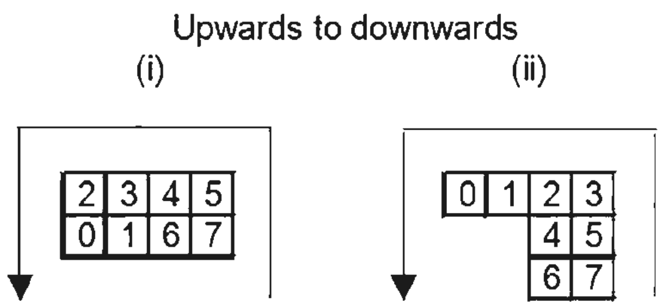


Figure 17 — Example of bit placement in (i) regular and (ii) irregular symbol characters when direction of placement changes

e) When the right-hand module column of the symbol character column encounters an alignment pattern or an area occupied by version information, bits are placed to form an irregular symbol character, extending along the single module column adjacent to the alignment pattern or version information. If the character ends before two columns are available for the next symbol character, the most significant bit of the next character shall be placed in the single column.

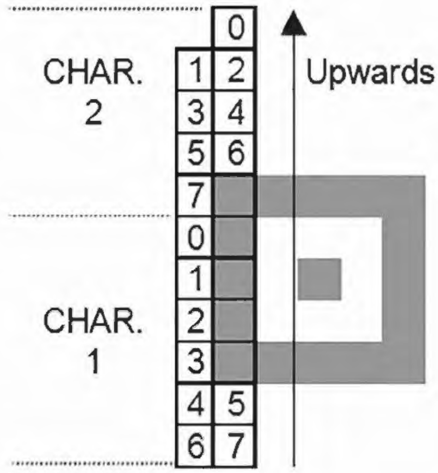


Figure 18 — Example of bit placement adjacent to alignment pattern

An alternative method for placement in the symbol, which yields the same result, is to regard the interleaved codeword sequence as a single bit stream, which is placed (starting with the most significant bit) in the two-module wide columns alternately upwards and downwards from the right to left of the symbol. In each column the bits are placed alternately in the right and left modules, moving upwards or downwards according to the direction of placement and skipping areas occupied by function patterns, changing direction at the top or bottom of the column. Each bit shall always be placed in the first available module position.

When the data capacity of the symbol is such that it does not divide exactly into a number of 8-bit symbol characters, the appropriate number of Remainder Bits (3, 4 or 7 as shown in Table 1) shall be used to fill the symbol capacity. These Remainder Bits shall always have the value 0 before data masking according to 7.8.

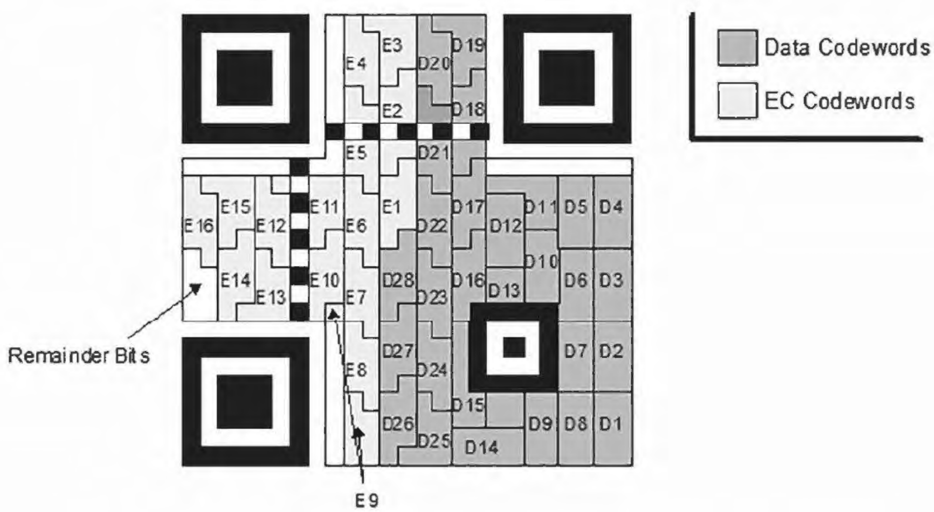
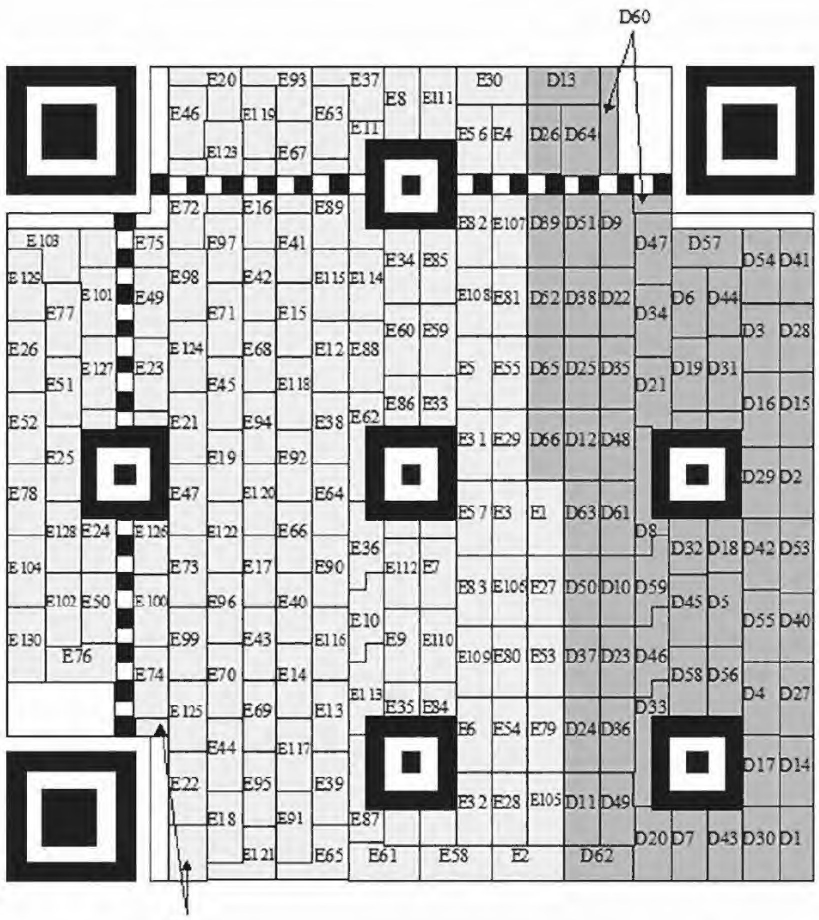


Figure 19 — Symbol character arrangement in version 2-M symbol



D 1 - D 13	Data Block 1
D 14 - D 26	Data Block 2
D 27 - D 39	Data Block 3
D 40 - D 52	Data Block 4
D 53 - D 66	Data Block 5
E 1 - E 26	EC Block 1
E 27 - E 52	EC Block 2
E 53 - E 78	EC Block 3
E 79 - E 104	EC Block 4
E 105 - E 130	EC Block 5

Figure 20 — Symbol character arrangement in version 7-H symbol

Exactly the same principles apply to a Micro QR Code symbol. There are no irregular symbol characters in these symbols and the sole exception is that D_3 in a Version M1 symbol, D_{11} in a Version M3-L symbol and D_9 in a Version M3-M symbol is a 2 × 2 square 4-module block.

7.8 Data masking

7.8.1 General

For reliable QR Code reading, it is preferable for dark and light modules to be arranged in a well-balanced manner in the symbol. The module pattern **1011101** particularly found in the finder pattern should be avoided in other areas of the symbol as much as possible. To meet the above conditions, data masking should be applied following the steps described below:

1. Data masking is not applied to function patterns.
2. Convert the given module pattern in the encoding region (excluding the format information and the version information) with multiple matrix patterns successively through the XOR operation. For the XOR operation, lay the module pattern over each of the data masking matrix patterns in turn and reverse the modules (from light to dark or vice versa) which correspond to dark modules of the data masking pattern.
3. Then evaluate all the resulting converted patterns by charging penalties for undesirable features on each conversion result.
4. Select the pattern with the lowest penalty points score.

7.8.2 Data mask patterns

Table 10 shows the data mask pattern reference (binary reference for use in the format information) and the data mask pattern generation condition. The data mask pattern is generated by defining as dark any module in the encoding region (excluding the area reserved for format information and the version information) for which the condition is true; in the condition, i refers to the row position of the module in question and j to its column position, with $(i, j) = (0, 0)$ for the top left module in the symbol.

Table 10 — Data mask pattern generation conditions

Data mask pattern reference for QR Code symbols	Data mask pattern reference for Micro QR Code symbols	Condition
000		$(i + j) \bmod 2 = 0$
001	00	$i \bmod 2 = 0$
010		$j \bmod 3 = 0$
011		$(i + j) \bmod 3 = 0$
100	01	$((i \operatorname{div} 2) + (j \operatorname{div} 3)) \bmod 2 = 0$
101		$(i\,j) \bmod 2 + (i\,j) \bmod 3 = 0$
110	10	$((i\,j) \bmod 2 + (i\,j) \bmod 3) \bmod 2 = 0$
111	11	$((i+j) \bmod 2 + (i\,j) \bmod 3) \bmod 2 = 0$

Figure 21 shows all data mask patterns, illustrated in a version 1 symbol. Figure 23 simulates the effects of data masking using data mask pattern references 000 to 111.

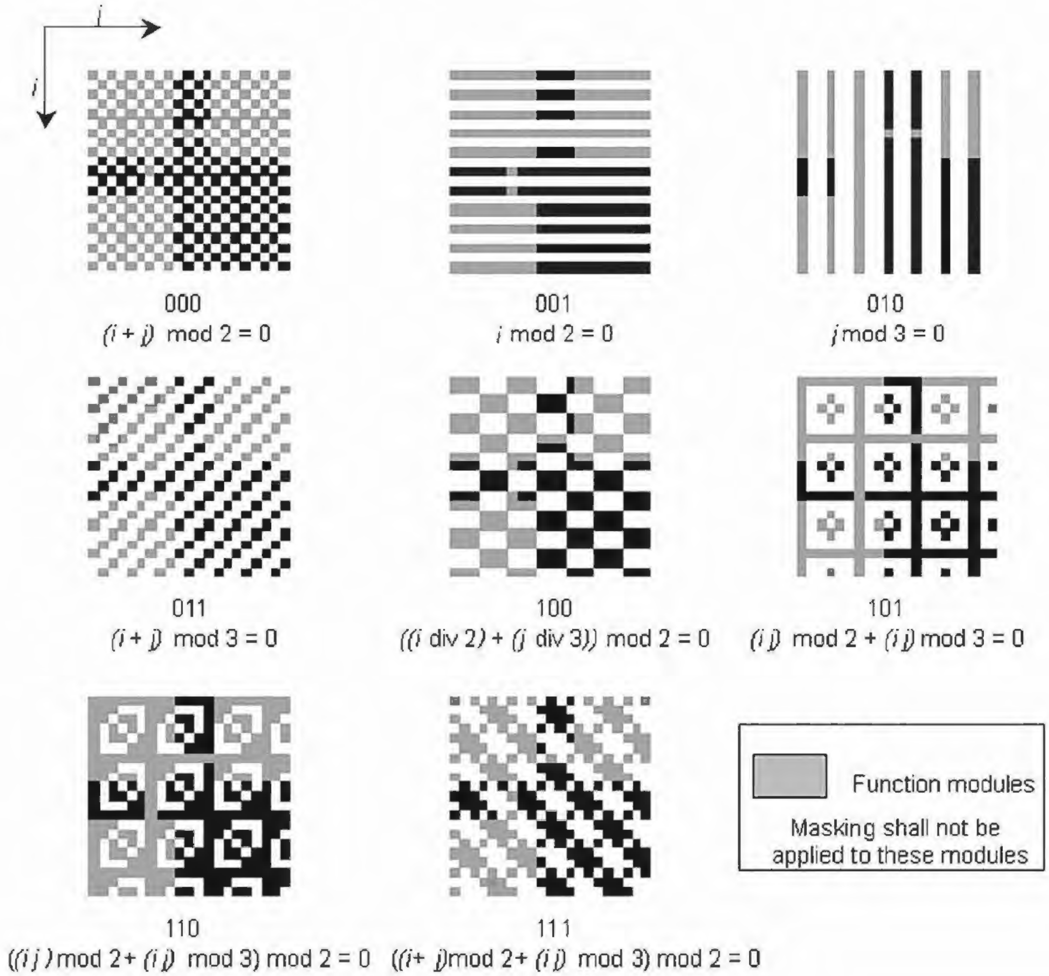


Figure 21 — Data mask patterns for version 1 symbol

NOTE 1 The three bits below each pattern represent the data mask pattern reference.

NOTE 2 The equation below the data mask pattern reference shows the data mask pattern generation condition; modules which meet the condition are shown dark.

Figure 22 below shows the four available data masking patterns applied to a Micro QR Code version M-4 symbol.

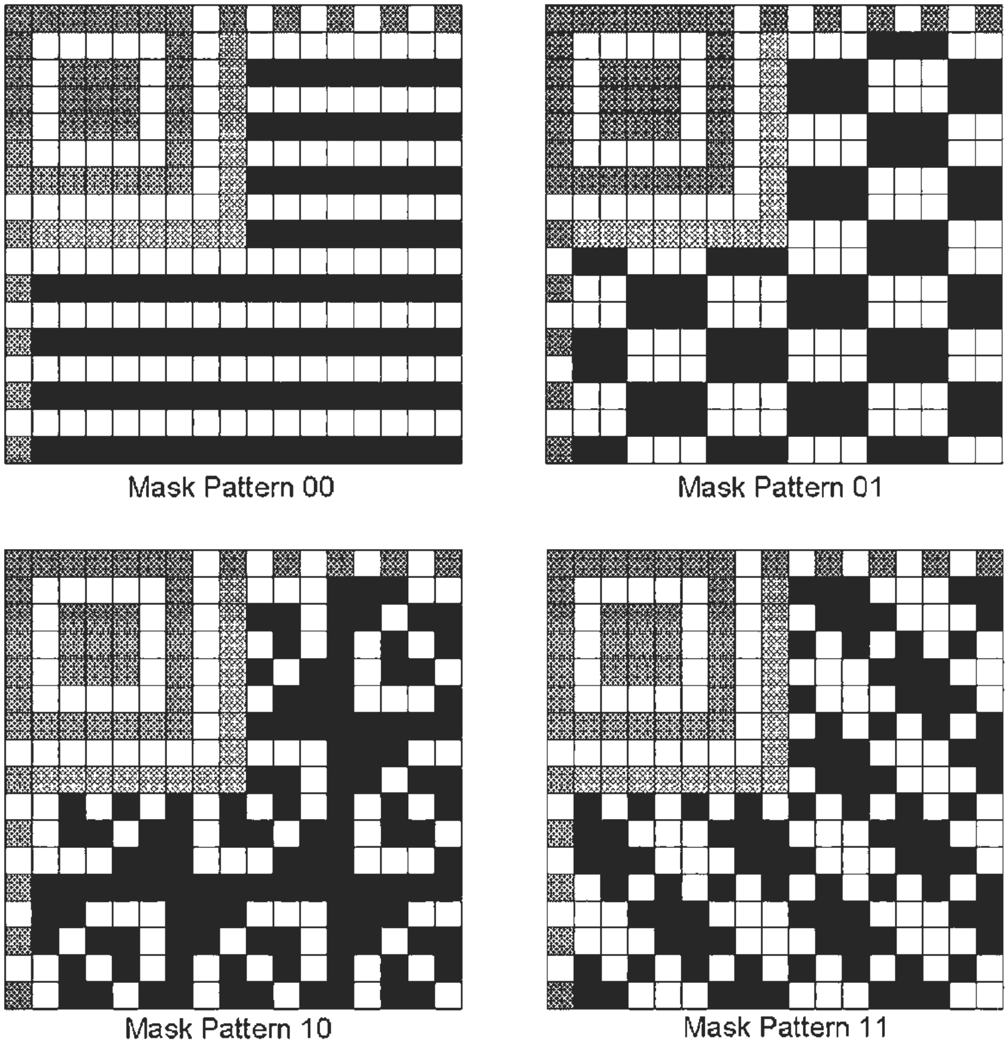


Figure 22 — Data mask patterns applied to Micro QR Code version M4 symbol

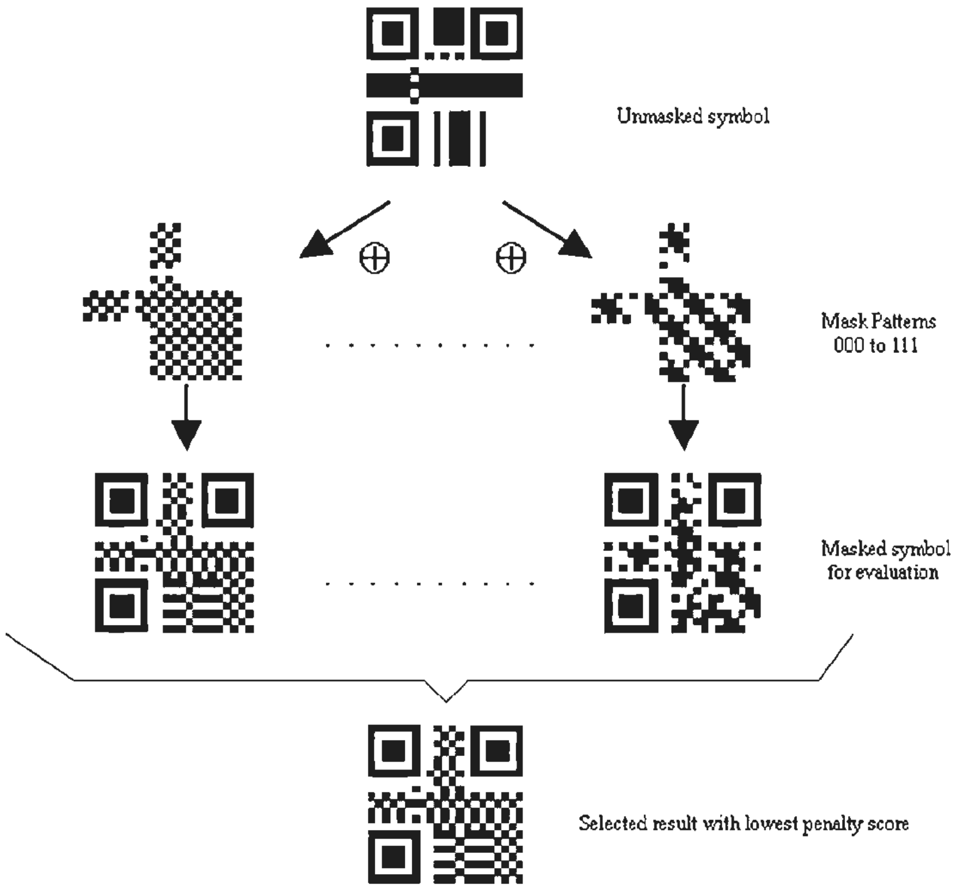


Figure 23 — Data masking simulation in QR Code symbols

7.8.3 Evaluation of data masking results

7.8.3.1 Evaluation of QR Code symbols

After performing the data masking operation with each data mask pattern in turn, the results shall be evaluated by scoring penalty points for each occurrence of the following features. The higher the number of points, the less acceptable the result. In Table 11 below, the variables N_1 to N_4 represent weighted penalty scores for the undesirable features ($N_1 = 3$, $N_2 = 3$, $N_3 = 40$, $N_4 = 10$), i is the amount by which the number of adjacent modules of the same color exceeds 5 and k is the rating of the deviation of the proportion of dark modules in the symbol from 50 % in steps of 5 %. Although the data masking operation is only performed on the encoding region of the symbol excluding the format information, the area to be evaluated is the complete symbol.

Table 11 — Scoring of data masking results

Feature	Evaluation condition	Points
Adjacent modules in row/column in same color	No. of modules = (5 + i)	$N_1 + i$
Block of modules in same color	Block size = 2 × 2	N_2
1 : 1 : 3 : 1 : 1 ratio (dark:light:dark:light:dark) pattern in row/column, preceded or followed by light area 4 modules wide	Existence of the pattern	N_3
Proportion of dark modules in entire symbol	50 × (5 × k)% to 50 × (5 × (k + 1))%	$N_4 \times k$

NOTE 1 Adjacent modules in row/column in the same colour.

Check the blocks consisting of light (white) or dark (black) modules of more than five in a row both laterally and vertically for the evaluation of data masking results. The rule of this calculation is that 3 penalty points shall be added to each block of five consecutive modules, 4 penalty points for each block of six consecutive modules and so on, with scoring by 1 point each time the number of modules increases. For example, impose 5 penalty points on the block of "dark:dark:dark:dark:dark:dar k:dark" module pattern, where a series of seven consecutive modules is counted as one block. However, do not double-count the point. The penalty point for a seven-module block, for example, shall be 5, not the sum of 3 (for a five-module block) + 4 (for a six-module block) + 5 (for a seven-module block) =12.

NOTE 2 Module blocks in the same colour.

The penalty point shall be equal to the number of blocks with 2 x 2 light or dark modules. Take a block consisting of 3 x 3 dark modules for an example. Considering that up to four 2 x 2 dark modules can be included in this block, the penalty applied to this block shall be calculated as 4 (blocks) x 3 (points) = 12 points.

NOTE 3 1:1:3:1:1 ratio pattern in row/column.

If the light area of more than 4 module wide exists after or before a 1:1:3:1:1 ratio (dark:light:dark:light:dark) pattern, the imposed penalty shall be 40 points.

NOTE 4 Proportion of dark modules in entire symbol.

Add 10 points to a deviation of 5% increment or decrement in the proportion ratio of dark module from the referential 50% (or 0 point) level. For example, assign 0 points as a penalty if the ratio of dark module is between 45% and 55%, or 10 points if the ratio of dark module is between 40% and 60%.

The data mask pattern which results in the lowest penalty score shall be selected for the symbol.

7.8.3.2 Evaluation of Micro QR Code symbols

After performing the data masking operation on the encoding region of the symbol with each data mask pattern in turn, the results shall be evaluated by scoring points for the number of dark modules in each of the two edges which are not timing patterns. The lower the number of points, the less acceptable the result. In these symbols, it is desirable to have more dark modules in the edge, in order to differentiate a quiet zone from an encoding region more effectively.

For each data mask pattern in turn, count the number of dark modules in the right and lower edges of the symbol (excluding the final module of the timing pattern). The evaluation score is given by the following formula:

If $SUM_1 \leq SUM_2$

$$\text{Evaluation score} = SUM_1 \times 16 + SUM_2$$

If $SUM_1 > SUM_2$

$$\text{Evaluation score} = SUM_2 \times 16 + SUM_1$$

where:

SUM_1 number of dark modules in right side edge

SUM_2 number of dark modules in lower side edge

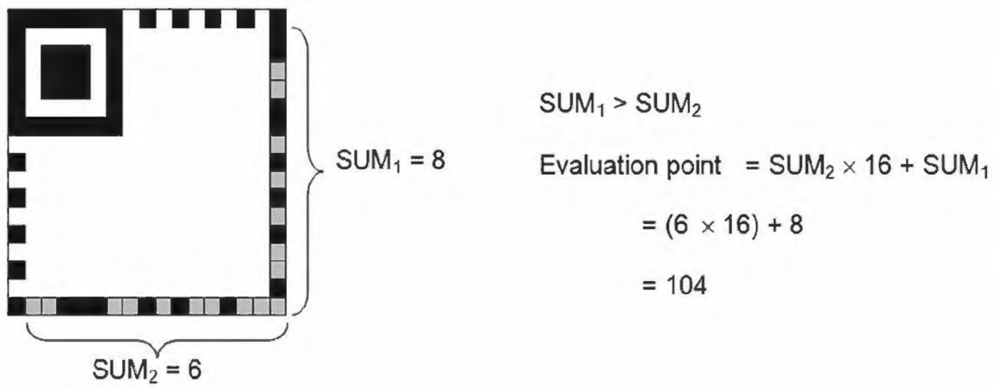


Figure 24 — Evaluation of masking results in Micro QR Code symbol

The data mask pattern which results in the highest score shall be selected for the symbol.

7.9 Format information

7.9.1 QR Code symbols

The format information is a 15-bit sequence containing 5 data bits, with 10 error correction bits calculated using the (15, 5) BCH code. For details of the error correction calculation for the format information, refer to Annex C. The first two data bits contain the Error Correction Level of the symbol, indicated in Table 12.

Table 12 — Error correction level indicators for QR Code symbols

Error Correction Level	Binary indicator
L	01
M	00
Q	11
H	10

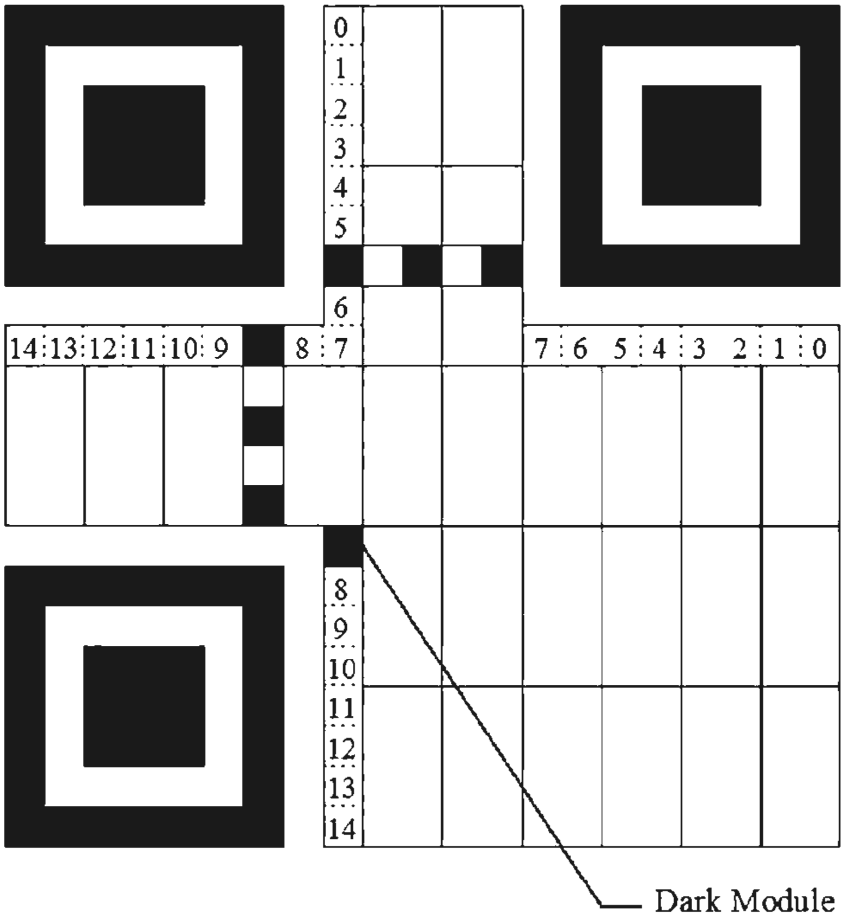
The third to fifth data bits of the format information contain the data mask pattern reference from Table 10 above for the pattern selected according to 7.8.3.

The 10 error correction bits shall be calculated as described in Annex C and appended to the 5 data bits.

The 15-bit error corrected format information shall then be XORed with the Mask Pattern **101010000010010**, in order to ensure that no combination of Error Correction Level and data mask pattern will result in an all-zero data string.

The resulting masked format information shall be mapped into the areas reserved for it in the symbol as shown in Figure 25. Note that the format information appears twice in the symbol in order to provide redundancy since its correct decoding is essential to the decoding of the complete symbol. The least

significant bit of the format information is located in the modules numbered 0, and the most significant bit in the modules numbered 14 in Figure 25. The module in position (4V + 9, 8) where V is the version number, shall always be dark and does not form part of the format information.



EXAMPLE

Assume Error Correction Level M	**00**
and data mask pattern reference:	**101**
Data:	**00101**
BCH bits:	**0011011100**
Unmasked bit sequence:	**001010011011100**
Mask pattern for XOR operation:	**101010000010010**
Format information module pattern:	**100000011001110**

bit 14 bit 0

Figure 25 — Format information positioning

7.9.2 Micro QR Code symbols

The format information is a 15-bit sequence containing 5 data bits, with 10 error correction bits calculated using the (15, 5) BCH code. For details of the error correction calculation for the format information, refer to Annex C. The first three data bits contain the symbol number (in binary), which identifies the version and error correction level, as shown in Table 13:

Table 13 — Symbol numbers for Micro QR Code symbols

Symbol number	Version	Error Correction Level	Binary Indicator
0	M1	Error detection only	000
1	M2	L	001
2	M2	M	010
3	M3	L	011
4	M3	M	100
5	M4	L	101
6	M4	M	110
7	M4	Q	111

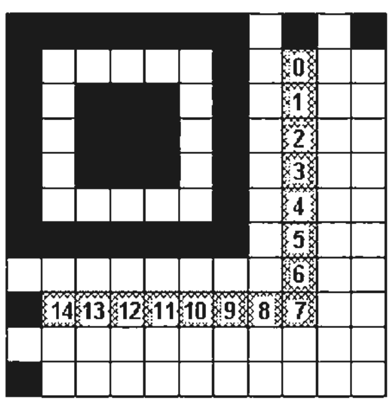
The fourth and fifth data bits of the format information contain the data mask pattern reference shown in Table 10 for the pattern selected according to 7.8.3.

The 10 error correction bits shall be calculated as described in Annex C and appended to the 5 data bits.

The 15-bit error corrected format information shall then be XORed with the bit pattern **100010001000101**, in order to ensure that no combination of symbol number and data mask pattern will result in an all-zero data string.

The resulting masked format information shall be mapped into the areas reserved for it in the symbol as shown in Figure 25 or 26, depending on the symbol type. The least significant bit of the format information is located in the module numbered 0, and the most significant bit in the module numbered 14 in Figures 24 and 25.

EXAMPLE



Symbol number 0: **000**

Data mask pattern reference:	**11**
Data bits (symbol number, data mask pattern reference):	**00011**
BCH bits:	**1101011001**
Unmasked bit sequence:	**000111101011001**
Mask pattern for XOR operation:	**100010001000101**
Format information module pattern:	**100101100011100**

bit 14 bit 0

Figure 26 — Micro QR Code symbol format information bit positions

7.10 Version information

The version information is included in QR Code symbols of version 7 or larger. It consists of an 18-bit sequence containing 6 data bits, with 12 error correction bits calculated using the (18, 6) Golay code. For details of the error correction calculation for the version information, refer to Annex D. The six data bits contain the Version of the symbol, most significant bit first.

The 12 error correction bits shall be calculated as described in Annex D and appended to the 6 data bits.

No version information will result in an all-zero data string since only Versions 7 to 40 symbols contain the version information. Masking is not therefore applied to the version information.

The resulting version information shall be mapped into the areas reserved for it in the symbol as shown in Figure 27. Note that the version information appears twice in the symbol in order to provide redundancy since its correct decoding is essential to the decoding of the complete symbol. The least significant bit of the version information is located in the modules numbered 0, and the most significant bit in the modules numbered 17, in Figure 28.

Example:

Version number:	**7**
Data:	**000111**
BCH bits:	**110010010100**
Version information module pattern:	**000111110010010100**

The version information areas are the 6 × 3 module block above the timing pattern and immediately to the left of the top right finder pattern separator, and the 3 × 6 module block to the left of the timing pattern and immediately above the lower left finder pattern separator.

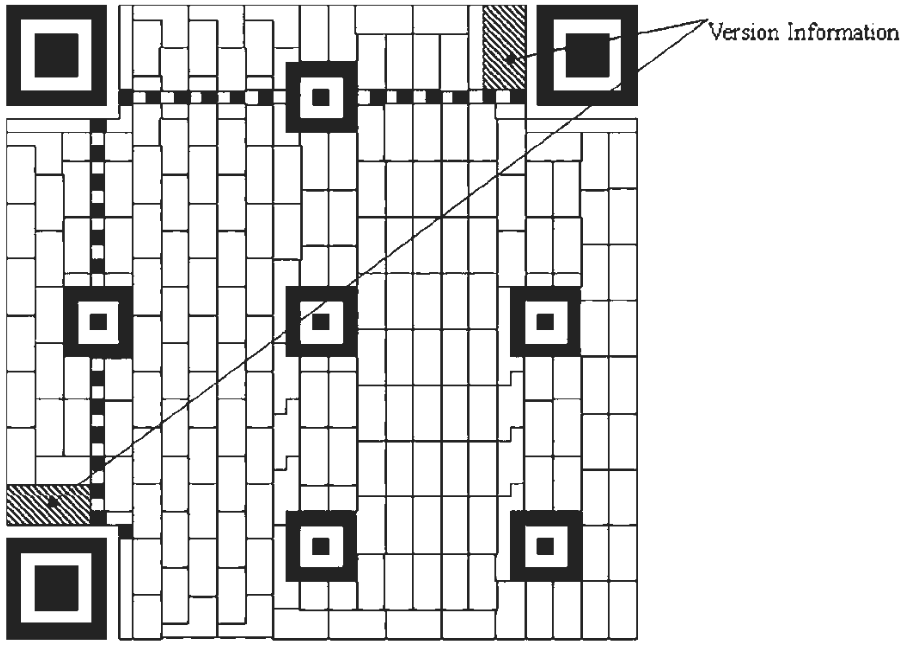


Figure 27 — Version information positioning

0	3	6	9	12	15
1	4	7	10	13	16
2	5	8	11	14	17

Version Information in lower left

0	1	2
3	4	5
6	7	8
9	10	11
12	13	14
15	16	17

Version Information in upper right

Figure 28 — Module arrangement in version information

8 Structured Append

8.1 Basic principles

Structured Append is not available with Micro QR Code symbols.

Up to 16 QR Code symbols may be appended in a structured format. If a symbol is part of a Structured Append message, it is indicated by a header block in the first two and a half symbol character positions.

The Structured Append mode indicator **0011** is placed in the four most significant bit positions in the first symbol character.

This is immediately followed by two Structured Append codewords, spread over the four least significant bits of the first symbol character, the second symbol character and the four most significant bits of the third symbol character. The first codeword is the symbol sequence indicator (see 7.2). The second codeword is the parity data (see 7.3) and is identical in all symbols in the message, enabling it to be verified that all symbols read form part of the same Structured Append message. This header is immediately followed by the data codewords for the symbol commencing with the first mode indicator. If one or more ECIs other than the default ECI is in force, an ECI header for each ECI, consisting of the ECI mode indicator and ECI Designator, shall follow the Structured Append header.

The lower part of Figure 29 shows an example of four Structured Append symbols, with the same data as that in the upper symbol.

Figure 29 — Single symbol (above) and Structured Append series of symbols (below) encoding "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"

8.2 Symbol Sequence Indicator

This codeword indicates the position of the symbol within the set of (up to 16) QR Code symbols in the Structured Append format (in the form *m* of *n* symbols). The first 4 bits of this codeword identify the position of the particular symbol. The last 4 bits identify the total number of symbols to be concatenated in the Structured Append format. The 4-bit patterns shall be the binary equivalents of (m - 1) and (n - 1) respectively.

EXAMPLE

To indicate the 3rd symbol of a set of 7, this shall be encoded thus:

3rd position:	**0010**
Total 7 symbols:	**0110**
Bit pattern:	**00100110**

8.3 Parity Data

The Parity Data shall be an 8-bit byte following the Symbol Sequence Indicator. The parity data is a value obtained by XORing byte by byte the byte values of all the original input data before division into symbol blocks. Mode Indicators, Character Count Identifiers, padding bits, Terminator and Pad Characters shall be excluded from the calculation. Input data is represented for this calculation by 2-byte Shift JIS values for Kanji (each byte being treated separately in the XOR calculation, most significant first) and 8-bit values as shown in Table 6 for other characters. In ECI mode the byte values obtained after any encryption or compression of the data shall be used for the calculation.

For example, "0123456789日本" is divided into "0123", "4567" and "89日本" as follows:

1st symbol block ("0123") - hex. values 30, 31, 32, 33

2nd symbol block ("4567") - hex. values 34, 35, 36, 37

3rd symbol block ("89日本") - hex. values 38, 39, 93FA, 967B

The parity data is calculated from "0123456789日本" by XORing the data successively, byte by byte.

$$30 \oplus 31 \oplus 32 \oplus 33 \oplus 34 \oplus 35 \oplus 36 \oplus 37 \oplus 38 \oplus 39 \oplus 93 \oplus \mathbf{FA} \oplus 96 \oplus \mathbf{7B} = \mathbf{85}$$

Note that the calculation of the parity data may be performed either before the data is sent to the printer or in the printer, based on the capabilities of the printer.

9 Symbol printing and marking

9.1 Dimensions

QR Code symbols shall conform to the following dimensions:

X dimension: the width of a module shall be specified by the application, taking into account the scanning technology to be used, and the technology to produce the symbol;

Y dimension: the height of a module shall be equal to the X dimension;

minimum quiet zone: equal to 2X (for Micro QR Code symbols) or 4X (for QR Code symbols) wide on all four sides.

9.2 Human-readable interpretation

Because QR Code symbols are capable of encoding thousands of characters, a human readable interpretation of the data characters may not be practical. As an alternative, descriptive text rather than literal text may accompany the symbol.

The character size and font are not specified, and the message may be printed anywhere in the area surrounding the symbol. The human readable interpretation should not interfere with the symbol itself nor the quiet zones.

9.3 Marking guidelines

QR Code symbols can be printed or marked using a number of different techniques. Annex K provides user guidelines.

10 Symbol quality

10.1 Methodology

QR Code symbols shall be assessed for quality using the 2D matrix bar code symbol print quality guidelines defined in ISO/IEC 15415, as augmented and modified below.

Some marking technologies may not be able to produce symbols conforming to this International Standard without taking special precautions. Annex M gives additional guidance to help any printing system achieve valid QR Code symbols.

Directly marked symbols (DPM) and/or symbols printed with disconnected dots may not pass this methodology and may not be readable by QR Code scanners. Application requiring such symbols should specify quality measurement using the ISO/IEC 15415 quality extension ISO/IEC/TR 29158 and may require specialized DPM scanners.

10.2 Symbol quality parameters

10.2.1 Fixed pattern damage

Annex G defines the measurement and grading basis for Fixed Pattern Damage.

10.2.2 Scan grade and overall symbol grade

The scan grade shall be the lowest of the grades for symbol contrast, modulation, fixed pattern damage, decode, axial non-uniformity, grid non-uniformity and unused error correction in an individual image of the symbol. The overall symbol grade is the arithmetic mean of the individual scan grades for a number of tested images of the symbol.

10.2.3 Grid non-uniformity

The ideal grid is calculated by using the finder patterns and alignment patterns as datum points, as located by the use of the reference decode algorithm (see Clause 12).

10.3 Process control measurements

A variety of tools and methods can be used to perform useful measurements for monitoring and controlling the process of creating QR Code symbols. These are described in Annex M. These techniques do not constitute a print quality check of the produced symbols (the method specified earlier in this clause and Annex G is the required method for assessing symbol print quality) but they individually and collectively yield good indications of whether the symbol print process is creating workable symbols.

11 Decoding procedure overview

The decoding steps from reading a QR Code symbol to outputting data characters are the reverse of the encoding procedure. Figure 30 shows an outline of the process flow.

1. Locate and obtain an image of the symbol. Recognize dark and light modules as an array of "0" and "1" bits. Identify reflectance polarity from finder pattern module colouring.

2. Read the format information. Release the masking pattern and perform error correction on the format information modules as necessary; if successful, symbol is in normal orientation, otherwise attempt mirror image decoding of format information. Identify Error Correction Level, either directly, in QR Code symbols, or from Micro QR Code symbol number, and data mask pattern reference.

3. Read the version information (where applicable), then determine the version of the symbol (from symbol number, in the case of Micro QR Code symbols).

4. Release the data masking by XORing the encoding region bit pattern with the data mask pattern the reference of which has been extracted from the format information.

5. Read the symbol characters according to the placement rules for the model and restore the data and error correction codewords of the message.

6. Detect errors using the error correction codewords corresponding to the Level Information. If any error is detected, correct it.

7. Divide the data codewords into segments according to the mode indicators and character count indicators.

8. Finally, decode the Data Characters in accordance with the mode(s) in use and output the result.

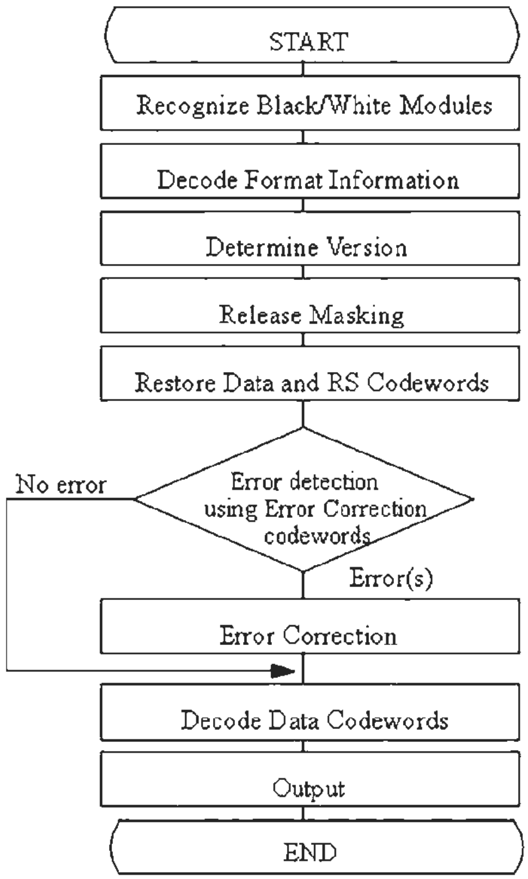


Figure 30 — QR Code decoding steps

12 Reference decode algorithm for QR Code

This reference decode algorithm finds the symbol in an image and decodes it. The decode algorithm refers to dark and light states in the image.

a) Determine a Global Threshold by taking a reflectance value midway between the maximum reflectance and minimum reflectance in the image. Convert the image to a set of dark and light pixels using the Global Threshold.

b) Locate the finder pattern. The finder pattern in QR Code consists of three identical finder patterns located at three of the four corners of the symbol. The finder pattern in Micro QR Code is a single finder pattern. As described in 6.3.3, module widths in each finder pattern form a dark-light-dark-light-dark sequence the relative widths of each element of which are in the ratios 1 : 1 : 3 : 1 : 1. For the purposes of this algorithm the tolerance for each of these widths is 0,5 (i.e. a range of 0,5 to 1,5 for the single module box and 2,5 to 3,5 for the three module square box).

1) When a candidate area is detected note the position of the first and last points A and B respectively at which a line of pixels in the image encounters the outer edges of the finder pattern (see Figure 31). Repeat this for adjacent pixel lines in the image until all lines crossing the central box of the finder pattern in the x axis of the image have been identified.

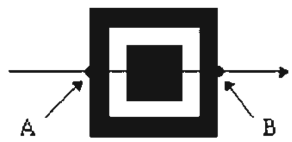


Figure 31 — Scan line in finder pattern

2) Repeat step 1) for pixel columns crossing the central box of the finder pattern in the y axis of the image.

3) Locate the center of the pattern. Construct a line through the midpoints between the points A and B on the outermost pixel lines crossing the central box of the finder pattern in the x axis. Construct a similar line through points A and B on the outermost pixel columns crossing the central box in the y axis. The center of the pattern is located at the intersection of these two lines.

4) Repeat steps 1) to 3) to locate the centers of the two other finder patterns.

5) If no candidate areas are detected, reverse the colouring of the light and dark pixels and recommence at the beginning of step b to attempt to decode the symbol as a symbol with reflectance reversal.

6) If a single pattern is identified but two further finder patterns cannot be located, attempt to decode the symbol as a Micro QR Code symbol by jumping to the Micro QR Code symbols reference decode (from step m).

c) Determine the rotational orientation of the symbol by analysing the finder pattern center coordinates to identify which pattern is the upper left pattern in the symbol and the angle of rotation of the symbol.

d) Determine 1) the distance D crossing the full width of the symbol between the centers of the upper left finder pattern and of the upper right finder pattern and 2) the width of the two patterns, W_{UL} and W_{UR} as shown in Figure 32.

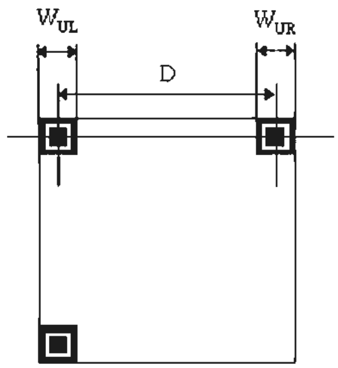


Figure 32 — Upper finder patterns

e) Calculate the nominal X dimension of the symbol.

$$X = (W_{UL} + W_{UR}) / 14$$

f) Provisionally determine the version V of the symbol.

$$V = \left[(D / X) - 10 \right] / 4$$

g) If the provisional symbol version is 6 or less, this is specified as the defined version. If the provisional symbol version is 7 or more, the version information is decoded as follows.

1) Divide the width W_{UR} of the upper right finder pattern by 7 to calculate the module size CP_{UR}.

$$CP_{UR} = W_{UR} / 7$$

2) Find the guide lines AC and AB from A, B and C, which pass through the centers of the three finder patterns, as shown in Figure 33 below. The sampling grid for each module center in the version information 1 area is determined based on lines parallel to the guide lines, the central coordinates of the finder patterns, and the module size CP_{UR}. Binary values 0 and 1 are determined from the light or dark pattern on the sampling grid.

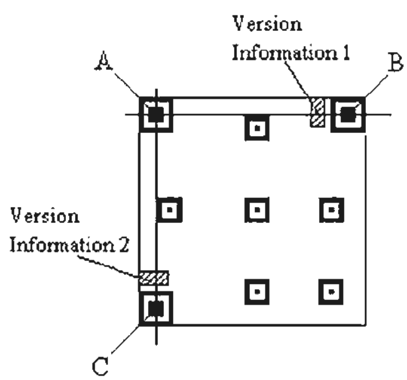


Figure 33 — Finder patterns and version information

3) Determine the version by detecting and correcting errors, if any, based on the table in Annex D.2.

4) If errors exceeding the error correction capacity are detected, then calculate the pattern width W_{DL} of the lower left finder pattern and follow a similar procedure to steps a), b) and c) above to decode version information 2.

h) For Version 1 symbols, redefine X as the average spacing of the center points of the dark and light modules in the upper side Timing Patterns. In a similar manner, calculate the Y dimension as the average spacing of the center points of the dark and light modules in the left side Timing Pattern. Establish a sampling grid based on 1) the horizontal line through the upper side Timing Pattern with lines parallel to it at the vertical spacing of Y, comprising six lines above the horizontal reference line and as many lines below it as are required for the version of the symbol and 2) the vertical line passing through the left side Timing Pattern with lines parallel to it at the horizontal spacing of X, comprising six lines to the left of the vertical reference line and as many lines to the right of it as are required for the version of the symbol. For version 2 and larger symbols, determine the central coordinate of each alignment pattern from the coordinates defined in 6.3.6 and Annex E and construct the sampling grids with lines equidistantly spaced between these points.

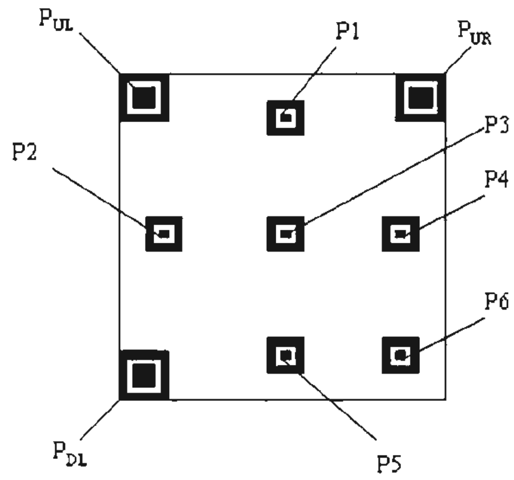


Figure 34 — Finder patterns and alignment patterns

1) Divide the pattern width W_{UL} of the upper left finder pattern P_{UL} by 7 to calculate the module size CP_{UL}.

$$CP_{UL} = W_{UL} / 7$$

2) Determine the provisional central coordinates of the alignment patterns P1 and P2 (see Figure 33), based on the coordinate of the center A of the upper left finder pattern P_{UL}, lines parallel to the guide lines AB and AC obtained in 7c), and the module size CP_{UL}.

3) Scan the outline of the white square in alignment pattern P1 and P2 starting from the pixel of the provisional central coordinate to find the actual central coordinates Xi and Yj (see Figure 35).

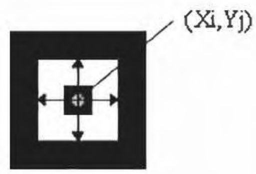


Figure 35 — Central coordinates of alignment pattern

4) Estimate the provisional central coordinate of the alignment pattern P3, based on the central coordinate of the upper left finder pattern P_{UL} and the actual central coordinates of the alignment patterns P1 and P2 obtained in step 3.

5) Find the actual central coordinate of the alignment pattern P3 by following the same procedure in step 3.

6) Find Lx, which is the center-to-center distance of the alignment patterns P2 and P3, and Ly, which is the center-to-center distance of the alignment patterns P1 and P3. Divide Lx and Ly by the defined spacing of the alignment patterns to obtain the module pitches CPx in the lower side and CPy in the right side in the upper left area of the symbol (see Figure 36).

CPx = Lx / AP

CPy = Ly / AP

where AP is the spacing in modules of the alignment pattern centers (see Table E.1).

In the same fashion, find Lx', which is the horizontal distance between the central coordinates of the upper left finder pattern P_{UL} and the central coordinates of the alignment pattern P1, and Ly', which is the vertical distance between the central coordinates of the upper left finder pattern P_{UL} and the central coordinates of the alignment pattern P2. Divide Lx' and Ly' by the formula below to obtain the module pitches CPx' in the upper side and CPy' in the left side in the upper left area of the symbol.

CPx' = Lx' / (Column coordinate of the central module of the alignment pattern P1

— Column coordinate of the central module of the upper left Finder Pattern P_{UL})

CPy' = Ly' / (Row coordinate of the central module of the alignment pattern P2

— Row coordinate of the central module of the upper left Finder Pattern P_{UL})

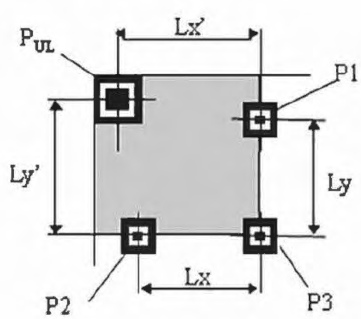


Figure 36 — Upper left area of symbol

7) Determine the sampling grid covering the upper left area of the symbol based on the module pitches CPx, CPx', CPy and CPy' representing each side in the upper left area of the symbol.

8) In the same fashion detemine the sampling grids for the upper right area (covered by the upper right finder pattern P_{UR}, alignment patterns P1, P3 and P4) and lower left area (covered by the lower left finder pattern P_{DL}, alignment patterns P2, P3 and P5) of the symbol.

9) For the alignment pattern P6 (see Figure 37), estimate its provisional central coordinate from the module pitches CPx' and CPy', the values of which are obtained from the spacings of alignment patterns P3, P4 and P5, guide lines passing through the centers of the alignment patterns P3 and P4, and P3 and P5 respectively, and the central coordinates of these Patterns.

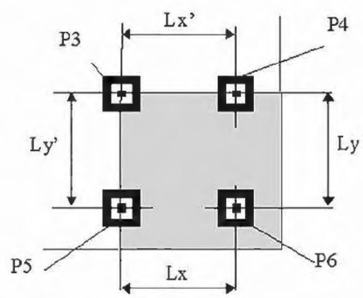


Figure 37 — Lower right area of symbol

10) Repeat steps 5) to 8) to determine the sampling grid for the lower right area of the symbol.

11) The same principles shall be applied to determine the sampling grids for any areas of the symbol not already covered.

i) Sample an area of 3 x 3 image pixels, centred on each intersection of the grid lines and determine whether it is dark or light based on the Global Threshold. Construct a bit matrix mapping the dark modules as binary 1 and light modules as binary 0.

j) Decode the format information adjacent to the upper left finder pattern as described in Annex C.3 to yield the Error Correction Level and the data mask pattern applied to the symbol. If errors exceeding the error correction capacity of the format information are detected, then follow the same procedure to decode the format information adjacent to the upper right and lower left finder patterns.

k) If a valid format information bit string cannot be derived, determine whether it is a valid sequence if read in the reverse direction and if so attempt to continue decoding as a mirror image symbol with the image row and column coordinates transposed.

l) Go to step y.

m) For Micro QR Code symbols, determine the possible angles of rotation of the symbol by analysing the angles of the lines from step b) 3) relative to the imaging sensor axes, as ϑ (see Figure 38), $\vartheta + 90°$, $\vartheta + 180°$ and $\vartheta + 270°$.

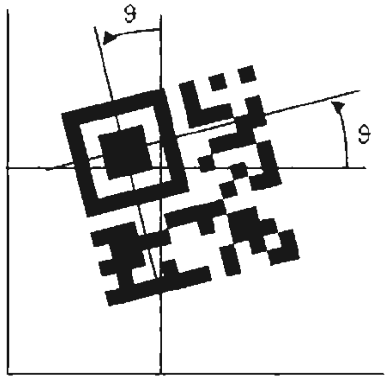


Figure 38 — angle ϑ relative to the imaging sensor axes

n) Plot three lines parallel to each axis of the finder pattern and equally spaced across the pattern and measure the distances from point A to point B on each line. The spacing is not limited but three lines shall be in the finder pattern.

o) Calculate the provisional module dimension X of the symbol in each axis as one seventh of the mean of the three distances A to B from step n.

p) Taking each side of the outer box of the finder pattern in turn, extend a line outward from the finder pattern in both directions, parallel to the edge and 0,5 X in from the edge.

q) Search for the timing patterns:

   1) Identify two edges of the finder pattern nominally perpendicular to each other, each of which has both

      i) a clear area of at least 1,5X in one direction;

      ii) alternating light and dark areas evenly spaced at 1X centres from the edge of the finder pattern in the opposite direction (a candidate timing pattern).

   2) Check that there is the same number of dark modules in each candidate timing pattern and that this number is between two and five.

r) Determine the provisional version of the symbol from the number of dark elements in the timing pattern:

   – If there are two dark elements, the symbol version is M1;

   – If there are three dark elements, the symbol version is M2;

   – If there are four dark elements, the symbol version is M3;

   – If there are five dark elements, the symbol version is M4.

s) From the centre of the first dark module in each side of the timing patterns extend a line parallel with the adjacent side of the finder pattern to intersect with the corresponding line from the other side and sample an area of 3 x 3 image pixels at 1X intervals along the line to determine the light or dark status of each module of the format information. Determine the format information bit string by taking the dark pixels as binary 1 and light pixels as binary 0.

t) Release masking of the format information by XORing the bit string with the pattern given in 7.9.2 and decode the format information (applying the error correction procedure given in Annex B if

necessary) to yield the symbol number (and hence the version and error correction level of the symbol) and the data mask pattern applied to the symbol.

u) If the format information bit string is not a valid sequence, determine whether it is a valid sequence if read in the reverse direction and if so attempt to continue decoding as a mirror image symbol with the image row and column coordinates transposed. If no more than two bits differ from a valid sequence in Annex C substitute this sequence and decode the substituted format information to obtain the symbol number and the data mask pattern.

v) Confirm the module pitch X in each axis by dividing the overall width from the outer edge of the finder pattern adjacent to the quiet zone to the outer edge of the last dark module in the timing pattern by the number of modules corresponding to the symbol version.

w) Establish a sampling grid, corresponding to the version of the symbol, of lines spaced 1X apart in each axis, parallel to each other and to the side of the finder pattern, and running from the centres of the timing pattern modules and from similar positions in the finder pattern.

x) Sample an area of 3 x 3 image pixels, centred on each intersection of the grid lines, and determine whether it is dark or light based on the Global Threshold. Construct a bit matrix mapping the dark modules as binary 1 and light modules as binary 0.

y) XOR the data mask pattern with the encoding region of the symbol to release the data masking and restore the symbol characters representing data and error correction codewords. This reverses the effect of the data masking process applied during the encoding procedure.

z) Determine the symbol codewords in accordance with the placement rules in 7.7.3.

  aa) Rearrange the codeword sequence into blocks as required for the symbol Version and Error Correction Level, by reversing the interleaving process defined in 7.6. step 3).

  bb) Follow the error detection and correction decoding procedure in Annex B to correct errors and erasures up to the maximum correction capacity for the symbol version and Error Correction Level.

  cc) Restore the original message bit stream by assembling the data blocks in sequence.

  dd) Subdivide the data bit stream into segments each commencing with a mode indicator and the length of which is determined by the character count indicator following the mode indicator.

  ee) Decode each segment according to the rules for the mode in force.

13 Autodiscrimination capability

QR Code can be used in an autodiscrimination environment with a number of other symbologies. (See Annex L). Although Model 1 and QR Code symbols can be autodiscriminated by analysis of the format information mask pattern, Model 1 symbols should not be used in the same environment as QR Code symbols.

14 Transmitted data

14.1 General principles

All encoded data characters shall be included in the data transmission. The function patterns, format and version information, error correction characters, Pad and Remainder characters shall not be transmitted. The default transmission mode for all data shall be as bytes.

The Structured Append header block shall not be transmitted by decoders operating in buffered mode which have reconstructed the complete message before transmission. If the decoder is operating in unbuffered mode the Structured Append header shall be transmitted as the first 2 bytes of every symbol.

More complex interpretations including the transmission of data in an Extended Channel Interpretation, are addressed below.

14.2 Symbology Identifier

ISO/IEC 15424 provides a standard procedure for reporting the symbology which has been read, together with options set in the decoder and any special features encountered in the symbol.

Once the structure of the data (including the use of any ECI) has been identified, the appropriate Symbology Identifier should be added by the decoder as a preamble to the transmitted data; if ECIs are used the Symbology Identifier is required. See Annex F for the Symbology Identifier and option values which apply to QR Code.

14.3 Extended Channel Interpretations

In systems where the ECI protocol is supported the transmission of the Symbology Identifier is required with every transmission. Whenever the ECI mode indicator is encountered, it shall be transmitted as the escape character $5C_{HEX}$, (which represents the backslash character "\" in ISO/IEC 8859-1 and in the AIM ECI specification and maps to the character "¥" in JIS X 0201). The codeword(s) representing the ECI Designator are converted into a 6 digit number by inverting the rules defined in Table 4. These 6 digits shall be transmitted as the corresponding 8-bit values in the range 30_{HEX} to 39_{HEX}, immediately following the escape character.

Application software recognizing **nnnnnn** should interpret all subsequent characters as being from the ECI defined by the 6 digit designator. This interpretation remains in effect until:

- the end of the encoded data;
- a change to a new ECI signaled by mode indicator **0111**, subject to rules defined by the AIM ECI specification.

When reverting to the default interpretation the decoder shall output the appropriate escape sequence as prefix to the data.

If the character $5C_{HEX}$ needs to be used as encoded data, transmission shall be as follows: whenever character $5C_{HEX}$ occurs as data, two bytes of the value shall be transmitted, thus a single occurrence is always an escape character and a double occurrence indicates true data.

Example 1

a) Encoded data (hex): 41 42 43 5C 31 32 33 34

Transmitted data: 41 42 43 5C 5C 31 32 33 34

b) Encoded data: ABC followed by <further data> encoded according to rules for ECI 123456.

Transmitted data: 41 42 43 5C 31 32 33 34 35 36 <further data>

Example 2 (using data in 7.4.2.2)

The message contains ECI mode indicator/ECI Designator/mode indicator/Character count indicator/Data in the form of

0111 00001001 0100 00000101 10100001 10100010 10100011 10100100 10100101

Symbology Identifier **]Q2** (see Annex F) must be added to the data transmission.

Transmission (hex. values): **5D 51 32 5C 30 30 30 30 30 39 A1 A2 A3 A4 A5**

Encoded data in ECI 000009: **ΑΒΓΔΕ**

In Structured Append mode, when the ECI mode indicator is encountered at the beginning of the symbol, subsequent data characters shall be interpreted as being from the ECI(s) in force at the end of the preceding symbol.

NOTE 5C$_{HEX}$ is equivalent to the backslash character "\" in ISO/IEC 8859-1 and to "¥" in JIS X 0201.

14.4 FNC1

In the modes with implied FNC1 in either first or second position, this implied character cannot be transmitted directly as there is no byte value corresponding to it. It is therefore necessary to indicate its presence in the first or second position by the transmission of the relevant Symbology Identifier (**]Q3,]Q4,]Q5** or **]Q6** defined in Annex F shall be used). Elsewhere in these symbols it may occur in accordance with the relevant application specification as a data field separator, represented in Alphanumeric mode by the character **%** and in Byte mode by the character **GS** (ASCII/JIS8 value 1D$_{HEX}$). In both cases the decoder shall transmit ASCII/JIS8 value 1D$_{HEX}$.

If, in symbols in FNC1 mode, the character **%** needs to be encoded as data while in Alphanumeric mode, it shall be represented in the symbol by **%%**. If this is encountered the decoder shall transmit a single **%** character.

Annex A
(normative)

Error detection and correction generator polynomials

The check character generation polynomial is used to divide the data codeword polynomial, where each codeword is the coefficient of the dividend polynomial in descending power order. The coefficients of the remainder of this division are the error correction codeword values.

Table A.1 shows the generator polynomials for the error correction codes which are used for each Version and Level, for all QR Code symbols. The number of error correction codewords required for a particular version and error correction level can be obtained from Table 9. In the table, α is the primitive element 2 under GF(2^8). Each generator polynomial is the product of the first degree polynomials: $x-2^0$, $x-2^1$, ..., $x-2^{n-1}$; where n is the degree of the generator polynomial.

Table A.1 — Generator polynomials for Reed-Solomon error correction codewords

Number of error correction codewords	Generator polynomials
2	$x^2 + \alpha^{25}x + \alpha$
5	$x^5 + \alpha^{113}x^4 + \alpha^{164}x^3 + \alpha^{166}x^2 + \alpha^{119}x + \alpha^{10}$
6	$x^6 + \alpha^{166}x^5 + x^4 + \alpha^{134}x^3 + \alpha^{5}x^2 + \alpha^{176}x + \alpha^{15}$
7	$x^7 + \alpha^{87}x^6 + \alpha^{229}x^5 + \alpha^{146}x^4 + \alpha^{149}x^3 + \alpha^{238}x^2 + \alpha^{102}x + \alpha^{21}$
8	$x^8 + \alpha^{175}x^7 + \alpha^{238}x^6 + \alpha^{208}x^5 + \alpha^{249}x^4 + \alpha^{215}x^3 + \alpha^{252}x^2 + \alpha^{196}x + \alpha^{28}$
10	$x^{10} + \alpha^{251}x^9 + \alpha^{67}x^8 + \alpha^{46}x^7 + \alpha^{61}x^6 + \alpha^{118}x^5 + \alpha^{70}x^4 + \alpha^{64}x^3 + \alpha^{94}x^2$ $+ \alpha^{32}x + \alpha^{45}$
13	$x^{13} + \alpha^{74}x^{12} + \alpha^{152}x^{11} + \alpha^{176}x^{10} + \alpha^{100}x^9 + \alpha^{86}x^8 + \alpha^{100}x^7 + \alpha^{106}x^6 + \alpha^{104}x^5$ $+ \alpha^{130}x^4 + \alpha^{218}x^3 + \alpha^{206}x^2 + \alpha^{140}x + \alpha^{78}$
14	$x^{14} + \alpha^{199}x^{13} + \alpha^{249}x^{12} + \alpha^{155}x^{11} + \alpha^{48}x^{10} + \alpha^{190}x^9 + \alpha^{124}x^8 + \alpha^{218}x^7$ $+ \alpha^{137}x^6 + \alpha^{216}x^5 + \alpha^{87}x^4 + \alpha^{207}x^3 + \alpha^{59}x^2 + \alpha^{22}x + \alpha^{91}$
15	$x^{15} + \alpha^{8}x^{14} + \alpha^{183}x^{13} + \alpha^{61}x^{12} + \alpha^{91}x^{11} + \alpha^{202}x^{10} + \alpha^{37}x^9 + \alpha^{51}x^8 + \alpha^{58}x^7$ $+ \alpha^{58}x^6 + \alpha^{237}x^5 + \alpha^{140}x^4 + \alpha^{124}x^3 + \alpha^{5}x^2 + \alpha^{99}x + \alpha^{105}$
16	$x^{16} + \alpha^{120}x^{15} + \alpha^{104}x^{14} + \alpha^{107}x^{13} + \alpha^{109}x^{12} + \alpha^{102}x^{11} + \alpha^{161}x^{10} + \alpha^{76}x^9$ $+ \alpha^{3}x^8 + \alpha^{91}x^7 + \alpha^{191}x^6 + \alpha^{147}x^5 + \alpha^{169}x^4\alpha^{182}x^3 + \alpha^{194}x^2 + \alpha^{225}x + \alpha^{120}$
17	$x^{17} + \alpha^{43}x^{16} + \alpha^{139}x^{15} + \alpha^{206}x^{14} + \alpha^{78}x^{13} + \alpha^{43}x^{12} + \alpha^{239}x^{11} + \alpha^{123}x^{10}$ $+ \alpha^{206}x^9 + \alpha^{214}x^8 + \alpha^{147}x^7 + \alpha^{24}x^6 + \alpha^{99}x^5 + \alpha^{150}x^4 + \alpha^{39}x^3 + \alpha^{243}x^2$ $+ \alpha^{163}x + \alpha^{136}$
18	$x^{18} + \alpha^{215}x^{17} + \alpha^{234}x^{16} + \alpha^{158}x^{15} + \alpha^{94}x^{14} + \alpha^{184}x^{13} + \alpha^{97}x^{12} + \alpha^{118}x^{11}$ $+ \alpha^{170}x^{10} + \alpha^{79}x^9 + \alpha^{187}x^8 + \alpha^{152}x^7 + \alpha^{148}x^6 + \alpha^{252}x^5 + \alpha^{179}x^4 + \alpha^{5}x^3$ $+ \alpha^{98}x^2 + \alpha^{96}x + \alpha^{153}$
20	$x^{20} + \alpha^{17}x^{19} + \alpha^{60}x^{18} + \alpha^{79}x^{17} + \alpha^{50}x^{16} + \alpha^{61}x^{15} + \alpha^{163}x^{14} + \alpha^{26}x^{13} + \alpha^{187}x^{12}$ $+ \alpha^{202}x^{11} + \alpha^{180}x^{10} + \alpha^{221}x^9 + \alpha^{225}x^8 + \alpha^{83}x^7 + \alpha^{239}x^6 + \alpha^{156}x^5 + \alpha^{164}x^4$ $+ \alpha^{212}x^3 + \alpha^{212}x^2 + \alpha^{188}x + \alpha^{190}$
22	$x^{22} + \alpha^{210}x^{21} + \alpha^{171}x^{20} + \alpha^{247}x^{19} + \alpha^{242}x^{18} + \alpha^{93}x^{17} + \alpha^{230}x^{16} + \alpha^{14}x^{15}$ $+ \alpha^{109}x^{14} + \alpha^{221}x^{13} + \alpha^{53}x^{12} + \alpha^{200}x^{11} + \alpha^{74}x^{10} + \alpha^{8}x^9 + \alpha^{172}x^8 + \alpha^{98}x^7$ $+ \alpha^{80}x^6 + \alpha^{219}x^5 + \alpha^{134}x^4 + \alpha^{160}x^3 + \alpha^{105}x^2 + \alpha^{165}x + \alpha^{231}$

Table A.1 *(continued)*

Number of error correction code-words	Generator polynomials
24	$x^{24} + \alpha^{229}x^{23} + \alpha^{121}x^{22} + \alpha^{135}x^{21} + \alpha^{48}x^{20} + \alpha^{211}x^{19} + \alpha^{117}x^{18} + \alpha^{251}x^{17} + \alpha^{126}x^{16} + \alpha^{159}x^{15} + \alpha^{180}x^{14} + \alpha^{169}x^{13} + \alpha^{152}x^{12} + \alpha^{192}x^{11} + \alpha^{226}x^{10} + \alpha^{228}x^{9} + \alpha^{218}x^{8} + \alpha^{111}x^{7} + x^{6} + \alpha^{117}x^{5} + \alpha^{232}x^{4} + \alpha^{87}x^{3} + \alpha^{96}x^{2} + \alpha^{227}x + \alpha^{21}$
26	$x^{26} + \alpha^{173}x^{25} + \alpha^{125}x^{24} + \alpha^{158}x^{23} + \alpha^{2}x^{22} + \alpha^{103}x^{21} + \alpha^{182}x^{20} + \alpha^{118}x^{19} + \alpha^{17}x^{18} + \alpha^{145}x^{17} + \alpha^{201}x^{16} + \alpha^{111}x^{15} + \alpha^{28}x^{14} + \alpha^{165}x^{13} + \alpha^{53}x^{12} + \alpha^{161}x^{11} + \alpha^{21}x^{10} + \alpha^{245}x^{9} + \alpha^{142}x^{8} + \alpha^{13}x^{7} + \alpha^{102}x^{6} + \alpha^{48}x^{5} + \alpha^{227}x^{4} + \alpha^{153}x^{3} + \alpha^{145}x^{2} + \alpha^{218}x + \alpha^{70}$
28	$x^{28} + \alpha^{168}x^{27} + \alpha^{223}x^{26} + \alpha^{200}x^{25} + \alpha^{104}x^{24} + \alpha^{224}x^{23} + \alpha^{234}x^{22} + \alpha^{108}x^{21} + \alpha^{180}x^{20} + \alpha^{110}x^{19} + \alpha^{190}x^{18} + \alpha^{195}x^{17} + \alpha^{147}x^{16} + \alpha^{205}x^{15} + \alpha^{27}x^{14} + \alpha^{232}x^{13} + \alpha^{201}x^{12} + \alpha^{21}x^{11} + \alpha^{43}x^{10} + \alpha^{245}x^{9} + \alpha^{87}x^{8} + \alpha^{42}x^{7} + \alpha^{195}x^{6} + \alpha^{212}x^{5} + \alpha^{119}x^{4} + \alpha^{242}x^{3} + \alpha^{37}x^{2} + \alpha^{9}x + \alpha^{123}$
30	$x^{30} + \alpha^{41}x^{29} + \alpha^{173}x^{28} + \alpha^{145}x^{27} + \alpha^{152}x^{26} + \alpha^{216}x^{25} + \alpha^{31}x^{24} + \alpha^{179}x^{23} + \alpha^{182}x^{22} + \alpha^{50}x^{21} + \alpha^{48}x^{20} + \alpha^{110}x^{19} + \alpha^{86}x^{18} + \alpha^{239}x^{17} + \alpha^{96}x^{16} + \alpha^{222}x^{15} + \alpha^{125}x^{14} + \alpha^{42}x^{13} + \alpha^{173}x^{12} + \alpha^{226}x^{11} + \alpha^{193}x^{10} + \alpha^{224}x^{9} + \alpha^{130}x^{8} + \alpha^{156}x^{7} + \alpha^{37}x^{6} + \alpha^{251}x^{5} + \alpha^{216}x^{4} + \alpha^{238}x^{3} + \alpha^{40}x^{2} + \alpha^{192}x + \alpha^{180}$
32	$x^{32}\alpha^{10}x^{31} + \alpha^{6}x^{30} + \alpha^{106}x^{29} + \alpha^{190}x^{28} + \alpha^{249}x^{27} + \alpha^{167}x^{26} + \alpha^{4}x^{25} + \alpha^{67}x^{24} + \alpha^{209}x^{23} + \alpha^{138}x^{22} + \alpha^{138}x^{21} + \alpha^{32}x^{20} + \alpha^{242}x^{19} + \alpha^{123}x^{18} + \alpha^{89}x^{17} + \alpha^{27}x^{16} + \alpha^{120}x^{15} + \alpha^{185}x^{14} + \alpha^{80}x^{13} + \alpha^{156}x^{12} + \alpha^{38}x^{11} + \alpha^{69}x^{10} + \alpha^{171}x^{9} + \alpha^{60}x^{8} + \alpha^{28}x^{7} + \alpha^{222}x^{6} + \alpha^{80}x^{5} + \alpha^{52}x^{4} + \alpha^{254}x^{3} + \alpha^{185}x^{2} + \alpha^{220}x + \alpha^{241}$
34	$x^{34} + \alpha^{111}x^{33} + \alpha^{77}x^{32} + \alpha^{146}x^{31} + \alpha^{94}x^{30} + \alpha^{26}x^{29} + \alpha^{21}x^{28} + \alpha^{108}x^{27} + \alpha^{19}x^{26} + \alpha^{105}x^{25} + \alpha^{94}x^{24} + \alpha^{113}x^{23} + \alpha^{193}x^{22} + \alpha^{86}x^{21} + \alpha^{140}x^{20} + \alpha^{163}x^{19} + \alpha^{125}x^{18} + \alpha^{58}x^{17} + \alpha^{158}x^{16} + \alpha^{229}x^{15} + \alpha^{239}x^{14} + \alpha^{218}x^{13} + \alpha^{103}x^{12} + \alpha^{56}x^{11} + \alpha^{70}x^{10} + \alpha^{114}x^{9} + \alpha^{61}x^{8} + \alpha^{183}x^{7} + \alpha^{129}x^{6} + \alpha^{167}x^{5} + \alpha^{13}x^{4} + \alpha^{98}x^{3} + \alpha^{62}x^{2} + \alpha^{129}x + \alpha^{51}$
36	$x^{36} + \alpha^{200}x^{35} + \alpha^{183}x^{34} + \alpha^{98}x^{33} + \alpha^{16}x^{32} + \alpha^{172}x^{31} + \alpha^{31}x^{30} + \alpha^{246}x^{29} + \alpha^{234}x^{28} + \alpha^{60}x^{27} + \alpha^{152}x^{26} + \alpha^{115}x^{25} + x^{24} + \alpha^{167}x^{23} + \alpha^{152}x^{22} + \alpha^{113}x^{21} + \alpha^{248}x^{20} + \alpha^{238}x^{19} + \alpha^{107}x^{18} + \alpha^{18}x^{17} + \alpha^{63}x^{16} + \alpha^{218}x^{15} + \alpha^{37}x^{14} + \alpha^{87}x^{13} + \alpha^{210}x^{12} + \alpha^{105}x^{11} + \alpha^{177}x^{10} + \alpha^{120}x^{9} + \alpha^{74}x^{8} + \alpha^{121}x^{7} + \alpha^{196}x^{6} + \alpha^{117}x^{5} + \alpha^{251}x^{4} + \alpha^{113}x^{3} + \alpha^{233}x^{2} + \alpha^{30}x + \alpha^{120}$
40	$x^{40} + \alpha^{59}x^{39} + \alpha^{116}x^{38} + \alpha^{79}x^{37} + \alpha^{161}x^{36} + \alpha^{252}x^{35} + \alpha^{98}x^{34} + \alpha^{128}x^{33} + \alpha^{205}x^{32} + \alpha^{128}x^{31} + \alpha^{161}x^{30} + \alpha^{247}x^{29} + \alpha^{57}x^{28} + \alpha^{163}x^{27} + \alpha^{56}x^{26} + \alpha^{235}x^{25} + \alpha^{106}x^{24} + \alpha^{53}x^{23} + \alpha^{26}x^{22} + \alpha^{187}x^{21} + \alpha^{174}x^{20} + \alpha^{226}x^{19} + \alpha^{104}x^{18} + \alpha^{170}x^{17} + \alpha^{7}x^{16} + \alpha^{175}x^{15} + \alpha^{35}x^{14} + \alpha^{181}x^{13} + \alpha^{114}x^{12} + \alpha^{88}x^{11} + \alpha^{41}x^{10} + \alpha^{47}x^{9} + \alpha^{163}x^{8} + \alpha^{125}x^{7} + \alpha^{134}x^{6} + \alpha^{72}x^{5} + \alpha^{20}x^{4} + \alpha^{232}x^{3} + \alpha^{53}x^{2} + \alpha^{35}x + \alpha^{15}$
42	$x^{42} + \alpha^{250}x^{41} + \alpha^{103}x^{40} + \alpha^{221}x^{39} + \alpha^{230}x^{38} + \alpha^{25}x^{37} + \alpha^{18}x^{36} + \alpha^{137}x^{35} + \alpha^{231}x^{34} + x^{33} + \alpha^{3}x^{32} + \alpha^{58}x^{31} + \alpha^{242}x^{30} + \alpha^{221}x^{29} + \alpha^{191}x^{28} + \alpha^{110}x^{27} + \alpha^{84}x^{26} + \alpha^{230}x^{25} + \alpha^{8}x^{24} + \alpha^{188}x^{23} + \alpha^{106}x^{22} + \alpha^{96}x^{21} + \alpha^{147}x^{20} + \alpha^{15}x^{19} + \alpha^{131}x^{18} + \alpha^{139}x^{17} + \alpha^{34}x^{16} + \alpha^{101}x^{15} + \alpha^{223}x^{14} + \alpha^{39}x^{13} + \alpha^{101}x^{12} + \alpha^{213}x^{11} + \alpha^{199}x^{10} + \alpha^{237}x^{9} + \alpha^{254}x^{8} + \alpha^{201}x^{7} + \alpha^{123}x^{6} + \alpha^{171}x^{5} + \alpha^{162}x^{4} + \alpha^{194}x^{3} + \alpha^{117}x^{2} + \alpha^{50}x + \alpha^{96}$
44	$x^{44} + \alpha^{190}x^{43} + \alpha^{7}x^{42} + \alpha^{61}x^{41} + \alpha^{121}x^{40} + \alpha^{71}x^{39} + \alpha^{246}x^{38} + \alpha^{69}x^{37} + \alpha^{55}x^{36} + \alpha^{168}x^{35} + \alpha^{188}x^{34} + \alpha^{89}x^{33} + \alpha^{243}x^{32} + \alpha^{191}x^{31} + \alpha^{25}x^{30} + \alpha^{72}x^{29} + \alpha^{123}x^{28} + \alpha^{9}x^{27} + \alpha^{145}x^{26} + \alpha^{14}x^{25} + \alpha^{247}x^{24} + \alpha x^{23} + \alpha^{238}x^{22} + \alpha^{44}x^{21} + \alpha^{78}x^{20} + \alpha^{143}x^{19} + \alpha^{62}x^{18} + \alpha^{224}x^{17} + \alpha^{126}x^{16} + \alpha^{118}x^{15} + \alpha^{114}x^{14} + \alpha^{68}x^{13} + \alpha^{163}x^{12} + \alpha^{52}x^{11} + \alpha^{194}x^{10} + \alpha^{217}x^{9} + \alpha^{147}x^{8} + \alpha^{204}x^{7} + \alpha^{169}x^{6} + \alpha^{37}x^{5} + \alpha^{130}x^{4} + \alpha^{113}x^{3} + \alpha^{102}x^{2} + \alpha^{73}x + \alpha^{181}$

Table A.1 *(continued)*

Number of error correction code-words	Generator polynomials
46	$x^{46} + \alpha^{112}x^{45} + \alpha^{94}x^{44} + \alpha^{88}x^{43} + \alpha^{112}x^{42} + \alpha^{253}x^{41} + \alpha^{224}x^{40} + \alpha^{202}x^{39} + \alpha^{115}x^{38} + \alpha^{187}x^{37} + \alpha^{99}x^{36} + \alpha^{89}x^{35} + \alpha^{5}x^{34} + \alpha^{54}x^{33} + \alpha^{113}x^{32} + \alpha^{129}x^{31} + \alpha^{44}x^{30} + \alpha^{58}x^{29} + \alpha^{16}x^{28} + \alpha^{135}x^{27} + \alpha^{216}x^{26} + \alpha^{169}x^{25} + \alpha^{211}x^{24} + \alpha^{36}x^{23} + \alpha x^{22} + \alpha^{4}x^{21} + \alpha^{96}x^{20} + \alpha^{60}x^{19} + \alpha^{241}x^{18} + \alpha^{73}x^{17} + \alpha^{104}x^{16} + \alpha^{234}x^{15} + \alpha^{8}x^{14} + \alpha^{249}x^{13} + \alpha^{245}x^{12} + \alpha^{119}x^{11} + \alpha^{174}x^{10} + \alpha^{52}x^{9} + \alpha^{25}x^{8} + \alpha^{157}x^{7} + \alpha^{224}x^{6} + \alpha^{43}x^{5} + \alpha^{202}x^{4} + \alpha^{223}x^{3} + \alpha^{19}x^{2} + \alpha^{82}x + \alpha^{15}$
48	$x^{48} + \alpha^{228}x^{47} + \alpha^{25}x^{46} + \alpha^{196}x^{45} + \alpha^{130}x^{44} + \alpha^{211}x^{43} + \alpha^{146}x^{42} + \alpha^{60}x^{41} + \alpha^{24}x^{40} + \alpha^{251}x^{39} + \alpha^{90}x^{38} + \alpha^{39}x^{37} + \alpha^{102}x^{36} + \alpha^{240}x^{35} + \alpha^{61}x^{34} + \alpha^{178}x^{33} + \alpha^{63}x^{32} + \alpha^{46}x^{31} + \alpha^{123}x^{30} + \alpha^{115}x^{29} + \alpha^{18}x^{28} + \alpha^{221}x^{27} + \alpha^{111}x^{26} + \alpha^{135}x^{25} + \alpha^{160}x^{24} + \alpha^{182}x^{23} + \alpha^{205}x^{22} + \alpha^{107}x^{21} + \alpha^{206}x^{20} + \alpha^{95}x^{19} + \alpha^{150}x^{18} + \alpha^{120}x^{17} + \alpha^{184}x^{16} + \alpha^{91}x^{15} + \alpha^{21}x^{14} + \alpha^{247}x^{13} + \alpha^{156}x^{12} + \alpha^{140}x^{11} + \alpha^{238}x^{10} + \alpha^{191}x^{9} + \alpha^{11}x^{8} + \alpha^{94}x^{7} + \alpha^{227}x^{6} + \alpha^{84}x^{5} + \alpha^{50}x^{4} + \alpha^{163}x^{3} + \alpha^{39}x^{2} + \alpha^{34}x + \alpha^{108}$
50	$x^{50} + \alpha^{232}x^{49} + \alpha^{125}x^{48} + \alpha^{157}x^{47} + \alpha^{161}x^{46} + \alpha^{164}x^{45} + \alpha^{9}x^{44} + \alpha^{118}x^{43} + \alpha^{46}x^{42} + \alpha^{209}x^{41} + \alpha^{99}x^{40} + \alpha^{203}x^{39} + \alpha^{193}x^{38} + \alpha^{35}x^{37} + \alpha^{3}x^{36} + \alpha^{209}x^{35} + \alpha^{111}x^{34} + \alpha^{195}x^{33} + \alpha^{242}x^{32} + \alpha^{203}x^{31} + \alpha^{225}x^{30} + \alpha^{46}x^{29} + \alpha^{13}x^{28} + \alpha^{32}x^{27} + \alpha^{160}x^{26} + \alpha^{126}x^{25} + \alpha^{209}x^{24} + \alpha^{130}x^{23} + \alpha^{160}x^{22} + \alpha^{242}x^{21} + \alpha^{215}x^{20} + \alpha^{242}x^{19} + \alpha^{75}x^{18} + \alpha^{77}x^{17} + \alpha^{42}x^{16} + \alpha^{189}x^{15} + \alpha^{32}x^{14} + \alpha^{113}x^{13} + \alpha^{65}x^{12} + \alpha^{124}x^{11} + \alpha^{69}x^{10} + \alpha^{228}x^{9} + \alpha^{114}x^{8} + \alpha^{235}x^{7} + \alpha^{175}x^{6} + \alpha^{124}x^{5} + \alpha^{170}x^{4} + \alpha^{215}x^{3} + \alpha^{232}x^{2} + \alpha^{133}x + \alpha^{205}$
52	$x^{52} + \alpha^{116}x^{51} + \alpha^{50}x^{50} + \alpha^{86}x^{49} + \alpha^{186}x^{48} + \alpha^{50}x^{47} + \alpha^{220}x^{46} + \alpha^{251}x^{45} + \alpha^{89}x^{44} + \alpha^{192}x^{43} + \alpha^{46}x^{42} + \alpha^{86}x^{41} + \alpha^{127}x^{40} + \alpha^{124}x^{39} + \alpha^{19}x^{38} + \alpha^{184}x^{37} + \alpha^{233}x^{36} + \alpha^{151}x^{35} + \alpha^{215}x^{34} + \alpha^{22}x^{33} + \alpha^{14}x^{32} + \alpha^{59}x^{31} + \alpha^{145}x^{30} + \alpha^{37}x^{29} + \alpha^{242}x^{28} + \alpha^{203}x^{27} + \alpha^{134}x^{26} + \alpha^{254}x^{25} + \alpha^{89}x^{24} + \alpha^{190}x^{23} + \alpha^{94}x^{22} + \alpha^{59}x^{21} + \alpha^{65}x^{20} + \alpha^{124}x^{19} + \alpha^{113}x^{18} + \alpha^{100}x^{17} + \alpha^{233}x^{16} + \alpha^{235}x^{15} + \alpha^{121}x^{14} + \alpha^{22}x^{13} + \alpha^{76}x^{12} + \alpha^{86}x^{11} + \alpha^{97}x^{10} + \alpha^{39}x^{9} + \alpha^{242}x^{8} + \alpha^{200}x^{7} + \alpha^{220}x^{6} + \alpha^{101}x^{5} + \alpha^{33}x^{4} + \alpha^{239}x^{3} + \alpha^{254}x^{2} + \alpha^{116}x + \alpha^{51}$
54	$x^{54} + \alpha^{183}x^{53} + \alpha^{26}x^{52} + \alpha^{201}x^{51} + \alpha^{87}x^{50} + \alpha^{210}x^{49} + \alpha^{221}x^{48} + \alpha^{113}x^{47} + \alpha^{21}x^{46} + \alpha^{46}x^{45} + \alpha^{65}x^{44} + \alpha^{45}x^{43} + \alpha^{50}x^{42} + \alpha^{238}x^{41} + \alpha^{184}x^{40} + \alpha^{249}x^{39} + \alpha^{225}x^{38} + \alpha^{102}x^{37} + \alpha^{58}x^{36} + \alpha^{209}x^{35} + \alpha^{218}x^{34} + \alpha^{109}x^{33} + \alpha^{165}x^{32} + \alpha^{26}x^{31} + \alpha^{95}x^{30} + \alpha^{184}x^{29} + \alpha^{192}x^{28} + \alpha^{52}x^{27} + \alpha^{245}x^{26} + \alpha^{35}x^{25} + \alpha^{254}x^{24} + \alpha^{238}x^{23} + \alpha^{175}x^{22} + \alpha^{172}x^{21} + \alpha^{79}x^{20} + \alpha^{123}x^{19} + \alpha^{25}x^{18} + \alpha^{122}x^{17} + \alpha^{43}x^{16} + \alpha^{120}x^{15} + \alpha^{108}x^{14} + \alpha^{215}x^{13} + \alpha^{80}x^{12} + \alpha^{128}x^{11} + \alpha^{201}x^{10} + \alpha^{235}x^{9} + \alpha^{8}x^{8} + \alpha^{153}x^{7} + \alpha^{59}x^{6} + \alpha^{101}x^{5} + \alpha^{31}x^{4} + \alpha^{198}x^{3} + \alpha^{76}x^{2} + \alpha^{31}x + \alpha^{156}$
56	$x^{56} + \alpha^{106}x^{55} + \alpha^{120}x^{54} + \alpha^{107}x^{53} + \alpha^{157}x^{52} + \alpha^{164}x^{51} + \alpha^{216}x^{50} + \alpha^{112}x^{49} + \alpha^{116}x^{48} + \alpha^{2}x^{47} + \alpha^{91}x^{46} + \alpha^{248}x^{45} + \alpha^{163}x^{44} + \alpha^{36}x^{43} + \alpha^{201}x^{42} + \alpha^{202}x^{41} + \alpha^{229}x^{40} + \alpha^{6}x^{39} + \alpha^{144}x^{38} + \alpha^{254}x^{37} + \alpha^{155}x^{36} + \alpha^{135}x^{35} + \alpha^{208}x^{34} + \alpha^{170}x^{33} + \alpha^{209}x^{32} + \alpha^{12}x^{31} + \alpha^{139}x^{30} + \alpha^{127}x^{29} + \alpha^{142}x^{28} + \alpha^{182}x^{27} + \alpha^{249}x^{26} + \alpha^{177}x^{25} + \alpha^{174}x^{24} + \alpha^{190}x^{23} + \alpha^{28}x^{22} + \alpha^{10}x^{21} + \alpha^{85}x^{20} + \alpha^{239}x^{19} + \alpha^{184}x^{18} + \alpha^{101}x^{17} + \alpha^{124}x^{16} + \alpha^{152}x^{15} + \alpha^{206}x^{14} + \alpha^{96}x^{13} + \alpha^{23}x^{12} + \alpha^{163}x^{11} + \alpha^{61}x^{10} + \alpha^{27}x^{9} + \alpha^{196}x^{8} + \alpha^{247}x^{7} + \alpha^{151}x^{6} + \alpha^{154}x^{5} + \alpha^{202}x^{4} + \alpha^{207}x^{3} + \alpha^{20}x^{2} + \alpha^{61}x + \alpha^{10}$
58	$x^{58} + \alpha^{82}x^{57} + \alpha^{116}x^{56} + \alpha^{26}x^{55} + \alpha^{247}x^{54} + \alpha^{66}x^{53} + \alpha^{27}x^{52} + \alpha^{62}x^{51} + \alpha^{107}x^{50} + \alpha^{252}x^{49} + \alpha^{182}x^{48} + \alpha^{200}x^{47} + \alpha^{185}x^{46} + \alpha^{235}x^{45} + \alpha^{55}x^{44} + \alpha^{251}x^{43} + \alpha^{242}x^{42} + \alpha^{210}x^{41} + \alpha^{144}x^{40} + \alpha^{154}x^{39} + \alpha^{237}x^{38} + \alpha^{176}x^{37} + \alpha^{141}x^{36} + \alpha^{192}x^{35} + \alpha^{248}x^{34} + \alpha^{152}x^{33} + \alpha^{249}x^{32} + \alpha^{206}x^{31} + \alpha^{85}x^{30} + \alpha^{253}x^{29} + \alpha^{142}x^{28} + \alpha^{65}x^{27} + \alpha^{165}x^{26} + \alpha^{125}x^{25} + \alpha^{23}x^{24} + \alpha^{24}x^{23} + \alpha^{30}x^{22} + \alpha^{122}x^{21} + \alpha^{240}x^{20} + \alpha^{214}x^{19} + \alpha^{6}x^{18} + \alpha^{129}x^{17} + \alpha^{218}x^{16} + \alpha^{29}x^{15} + \alpha^{145}x^{14} + \alpha^{127}x^{13} + \alpha^{134}x^{12} + \alpha^{206}x^{11} + \alpha^{245}x^{10} + \alpha^{117}x^{9} + \alpha^{29}x^{8} + \alpha^{41}x^{7} + \alpha^{63}x^{6} + \alpha^{159}x^{5} + \alpha^{142}x^{4} + \alpha^{233}x^{3} + \alpha^{125}x^{2} + \alpha^{148}x + \alpha^{123}$

Table A.1 *(continued)*

Number of error correction code-words	Generator polynomials
60	$x^{60} + \alpha^{107}x^{59} + \alpha^{140}x^{58} + \alpha^{26}x^{57} + \alpha^{12}x^{56} + \alpha^{9}x^{55} + \alpha^{141}x^{54} + \alpha^{243}x^{53} + \alpha^{197}x^{52} + \alpha^{226}x^{51} + \alpha^{197}x^{50} + \alpha^{219}x^{49} + \alpha^{45}x^{48} + \alpha^{211}x^{47} + \alpha^{101}x^{46} + \alpha^{219}x^{45} + \alpha^{120}x^{44} + \alpha^{28}x^{43} + \alpha^{181}x^{42} + \alpha^{127}x^{41} + \alpha^{6}x^{40} + \alpha^{100}x^{39} + \alpha^{247}x^{38} + \alpha^{2}x^{37} + \alpha^{205}x^{36} + \alpha^{198}x^{35} + \alpha^{57}x^{34} + \alpha^{115}x^{33} + \alpha^{219}x^{32} + \alpha^{101}x^{31} + \alpha^{109}x^{30} + \alpha^{160}x^{29} + \alpha^{82}x^{28} + \alpha^{37}x^{27} + \alpha^{38}x^{26} + \alpha^{238}x^{25} + \alpha^{49}x^{24} + \alpha^{160}x^{23} + \alpha^{209}x^{22} + \alpha^{121}x^{21} + \alpha^{86}x^{20} + \alpha^{11}x^{19} + \alpha^{124}x^{18} + \alpha^{30}x^{17} + \alpha^{181}x^{16} + \alpha^{84}x^{15} + \alpha^{25}x^{14} + \alpha^{194}x^{13} + \alpha^{87}x^{12} + \alpha^{65}x^{11} + \alpha^{102}x^{10} + \alpha^{190}x^{9} + \alpha^{220}x^{8} + \alpha^{70}x^{7} + \alpha^{27}x^{6} + \alpha^{209}x^{5} + \alpha^{16}x^{4} + \alpha^{89}x^{3} + \alpha^{7}x^{2} + \alpha^{33}x + \alpha^{240}$
62	$x^{62} + \alpha^{65}x^{61} + \alpha^{202}x^{60} + \alpha^{113}x^{59} + \alpha^{98}x^{58} + \alpha^{71}x^{57} + \alpha^{223}x^{56} + \alpha^{248}x^{55} + \alpha^{118}x^{54} + \alpha^{214}x^{53} + \alpha^{94}x^{52} + x^{51} + \alpha^{122}x^{50} + \alpha^{37}x^{49} + \alpha^{23}x^{48} + \alpha^{2}x^{47} + \alpha^{228}x^{46} + \alpha^{58}x^{45} + \alpha^{121}x^{44} + \alpha^{7}x^{43} + \alpha^{105}x^{42} + \alpha^{135}x^{41} + \alpha^{78}x^{40} + \alpha^{243}x^{39} + \alpha^{118}x^{38} + \alpha^{70}x^{37} + \alpha^{76}x^{36} + \alpha^{223}x^{35} + \alpha^{89}x^{34} + \alpha^{72}x^{33} + \alpha^{50}x^{32} + \alpha^{70}x^{31} + \alpha^{111}x^{30} + \alpha^{194}x^{29} + \alpha^{17}x^{28} + \alpha^{212}x^{27} + \alpha^{126}x^{26} + \alpha^{181}x^{25} + \alpha^{35}x^{24} + \alpha^{221}x^{23} + \alpha^{117}x^{22} + \alpha^{235}x^{21} + \alpha^{11}x^{20} + \alpha^{229}x^{19} + \alpha^{149}x^{18} + \alpha^{147}x^{17} + \alpha^{123}x^{16} + \alpha^{213}x^{15} + \alpha^{40}x^{14} + \alpha^{115}x^{13} + \alpha^{6}x^{12} + \alpha^{200}x^{11} + \alpha^{100}x^{10} + \alpha^{26}x^{9} + \alpha^{246}x^{8} + \alpha^{182}x^{7} + \alpha^{218}x^{6} + \alpha^{127}x^{5} + \alpha^{215}x^{4} + \alpha^{36}x^{3} + \alpha^{186}x^{2} + \alpha^{110}x + \alpha^{106}$
64	$x^{64} + \alpha^{45}x^{63} + \alpha^{51}x^{62} + \alpha^{175}x^{61} + \alpha^{9}x^{60} + \alpha^{7}x^{59} + \alpha^{158}x^{58} + \alpha^{159}x^{57} + \alpha^{49}x^{56} + \alpha^{68}x^{55} + \alpha^{119}x^{54} + \alpha^{92}x^{53} + \alpha^{123}x^{52} + \alpha^{177}x^{51} + \alpha^{204}x^{50} + \alpha^{187}x^{49} + \alpha^{254}x^{48} + \alpha^{200}x^{47} + \alpha^{78}x^{46} + \alpha^{141}x^{45} + \alpha^{149}x^{44} + \alpha^{119}x^{43} + \alpha^{26}x^{42} + \alpha^{127}x^{41} + \alpha^{53}x^{40} + \alpha^{160}x^{39} + \alpha^{93}x^{38} + \alpha^{199}x^{37} + \alpha^{212}x^{36} + \alpha^{29}x^{35} + \alpha^{24}x^{34} + \alpha^{145}x^{33} + \alpha^{156}x^{32} + \alpha^{208}x^{31} + \alpha^{150}x^{30} + \alpha^{218}x^{29} + \alpha^{209}x^{28} + \alpha^{4}x^{27} + \alpha^{216}x^{26} + \alpha^{91}x^{25} + \alpha^{47}x^{24} + \alpha^{184}x^{23} + \alpha^{146}x^{22} + \alpha^{47}x^{21} + \alpha^{140}x^{20} + \alpha^{195}x^{19} + \alpha^{195}x^{18} + \alpha^{125}x^{17} + \alpha^{242}x^{16} + \alpha^{238}x^{15} + \alpha^{63}x^{14} + \alpha^{99}x^{13} + \alpha^{108}x^{12} + \alpha^{140}x^{11} + \alpha^{230}x^{10} + \alpha^{242}x^{9} + \alpha^{31}x^{8} + \alpha^{204}x^{7} + \alpha^{11}x^{6} + \alpha^{178}x^{5} + \alpha^{243}x^{4} + \alpha^{217}x^{3} + \alpha^{156}x^{2} + \alpha^{213}x + \alpha^{231}$
66	$x^{66} + \alpha^{5}x^{65} + \alpha^{118}x^{64} + \alpha^{222}x^{63} + \alpha^{180}x^{62} + \alpha^{136}x^{61} + \alpha^{136}x^{60} + \alpha^{162}x^{59} + \alpha^{51}x^{58} + \alpha^{46}x^{57} + \alpha^{117}x^{56} + \alpha^{13}x^{55} + \alpha^{215}x^{54} + \alpha^{81}x^{53} + \alpha^{17}x^{52} + \alpha^{139}x^{51} + \alpha^{247}x^{50} + \alpha^{197}x^{49} + \alpha^{171}x^{48} + \alpha^{95}x^{47} + \alpha^{173}x^{46} + \alpha^{65}x^{45} + \alpha^{137}x^{44} + \alpha^{178}x^{43} + \alpha^{68}x^{42} + \alpha^{111}x^{41} + \alpha^{95}x^{40} + \alpha^{101}x^{39} + \alpha^{41}x^{38} + \alpha^{72}x^{37} + \alpha^{214}x^{36} + \alpha^{169}x^{35} + \alpha^{197}x^{34} + \alpha^{95}x^{33} + \alpha^{7}x^{32} + \alpha^{44}x^{31} + \alpha^{154}x^{30} + \alpha^{77}x^{29} + \alpha^{111}x^{28} + \alpha^{236}x^{27} + \alpha^{40}x^{26} + \alpha^{121}x^{25} + \alpha^{143}x^{24} + \alpha^{63}x^{23} + \alpha^{87}x^{22} + \alpha^{80}x^{21} + \alpha^{253}x^{20} + \alpha^{240}x^{19} + \alpha^{126}x^{18} + \alpha^{217}x^{17} + \alpha^{77}x^{16} + \alpha^{34}x^{15} + \alpha^{232}x^{14} + \alpha^{106}x^{13} + \alpha^{50}x^{12} + \alpha^{168}x^{11} + \alpha^{82}x^{10} + \alpha^{76}x^{9} + \alpha^{146}x^{8} + \alpha^{67}x^{7} + \alpha^{106}x^{6} + \alpha^{171}x^{5} + \alpha^{25}x^{4} + \alpha^{132}x^{3} + \alpha^{93}x^{2} + \alpha^{45}x + \alpha^{105}$
68	$x^{68} + \alpha^{247}x^{67} + \alpha^{159}x^{66} + \alpha^{223}x^{65} + \alpha^{33}x^{64} + \alpha^{224}x^{63} + \alpha^{93}x^{62} + \alpha^{77}x^{61} + \alpha^{70}x^{60} + \alpha^{90}x^{59} + \alpha^{160}x^{58} + \alpha^{32}x^{57} + \alpha^{254}x^{56} + \alpha^{43}x^{55} + \alpha^{150}x^{54} + \alpha^{84}x^{53} + \alpha^{101}x^{52} + \alpha^{190}x^{51} + \alpha^{205}x^{50} + \alpha^{133}x^{49} + \alpha^{52}x^{48} + \alpha^{60}x^{47} + \alpha^{202}x^{46} + \alpha^{165}x^{45} + \alpha^{220}x^{44} + \alpha^{203}x^{43} + \alpha^{151}x^{42} + \alpha^{93}x^{41} + \alpha^{84}x^{40} + \alpha^{15}x^{39} + \alpha^{84}x^{38} + \alpha^{253}x^{37} + \alpha^{173}x^{36} + \alpha^{160}x^{35} + \alpha^{89}x^{34} + \alpha^{227}x^{33} + \alpha^{52}x^{32} + \alpha^{199}x^{31} + \alpha^{97}x^{30} + \alpha^{95}x^{29} + \alpha^{231}x^{28} + \alpha^{52}x^{27} + \alpha^{177}x^{26} + \alpha^{41}x^{25} + \alpha^{125}x^{24} + \alpha^{137}x^{23} + \alpha^{241}x^{22} + \alpha^{166}x^{21} + \alpha^{225}x^{20} + \alpha^{118}x^{19} + \alpha^{2}x^{18} + \alpha^{54}x^{17} + \alpha^{32}x^{16} + \alpha^{82}x^{15} + \alpha^{215}x^{14} + \alpha^{175}x^{13} + \alpha^{198}x^{12} + \alpha^{43}x^{11} + \alpha^{238}x^{10} + \alpha^{235}x^{9} + \alpha^{27}x^{8} + \alpha^{101}x^{7} + \alpha^{184}x^{6} + \alpha^{127}x^{5} + \alpha^{3}x^{4} + \alpha^{5}x^{3} + \alpha^{8}x^{2} + \alpha^{163}x + \alpha^{238}$

Annex B
(normative)

Error correction decoding steps

Take the Version 1-M symbol as an example. For the symbol, the (26, 16, 4) Reed-Solomon code under GF(2^8) is used for error correction. Provided that the code after releasing data masking from the symbol is:

$$R = (r_0, r_1, r_2, \ldots, r_{25})$$

That is,

$$R(x) = r_0 + r_1 x + r_2 x^2 + \ \ldots \ + r_{25} x^{25}$$

r_i(i=0-25) is an element of GF(2^8)

(i) Calculate n syndromes (where n is equal to the number of codewords available for error correction, given by ($c - k - p$) as shown in Table 9).

Find the syndrome S_i(i=0-(n – 1)).

$$S_0 = R(1) \ = r_0 + r_1 + r_2 + \ \ldots \ + r_{25}$$
$$S_1 = R(\alpha) \ = r_0 + r_1\alpha + r_2\alpha^2 + \ \ldots \ + r_{25}\alpha^{25}$$
$$\ldots$$
$$\ldots$$
$$S_7 = R(\alpha^7) \ = r_0 + r_1\alpha^7 + r_2\alpha^{14} + \ \ldots \ + r_{25}\alpha^{175}$$

where α is a primitive element of GF(2^8)

(ii) Find the error positions:

$$S_0\sigma_4 - S_1\sigma_3 + S_2\sigma_2 - S_3\sigma_1 + S_4 = 0$$
$$S_1\sigma_4 - S_2\sigma_3 + S_3\sigma_2 - S_4\sigma_1 + S_5 = 0$$
$$S_2\sigma_4 - S_3\sigma_3 + S_4\sigma_2 - S_5\sigma_1 + S_6 = 0$$
$$S_3\sigma_4 - S_4\sigma_3 + S_5\sigma_2 - S_6\sigma_1 + S_7 = 0$$

Find the variable σ_i(i = 1-4) for each error position using the above formulas. Then, substitute the variable for the following polynomial and substitute elements of GF(2^8) one by one.

$$\sigma(x) \ = \ \sigma_4 + \ \sigma_3 x + \ \sigma_2 x^2 + \ \sigma_1 x^3 + x^4$$

Now, it is found that an error is on the jth digit (counting from the 0-th digit) for the element αj which makes σ (α) = 0.

(iii) Find the error size.

Supposing that an error is on the $j1, j2, j4$ digits in (ii) above, then find the size of the error.

$$Y_1\alpha j^1 + Y_2\alpha j^2 + Y_3\alpha j^3 + Y_4\alpha j^4 = S_0$$
$$Y_1\alpha^2 j^1 + Y_2\alpha^2 j^2 + Y_3\alpha^2 j^3 + Y_4\alpha^2 j^4 = S_1$$
$$Y_1\alpha^3 j^1 + Y_2\alpha^3 j^2 + Y_3\alpha^3 j^3 + Y_4\alpha^3 j^4 = S_2$$
$$Y_1\alpha^4 j^1 + Y_2\alpha^4 j^2 + Y_3\alpha^4 j^3 + Y_4\alpha^4 j^4 = S_3$$

Solve the above equations to find the size of each error $Y_i (i = 1\text{-}4)$.

(iv) Correct the error.

Correct the error by adding the complement of the error size value to each error position.

Annex C
(normative)

Format information

C.1 General

The format information consists of a 15-bit sequence comprising 5 data bits and 10 BCH error correction bits. This Annex describes the calculation of the error correction bits and the error correction decoding process.

C.2 Error correction bit calculation

The Bose-Chaudhuri-Hocquenghem (15,5) code shall be used for error correction. The polynomial whose coefficient is the data bit string shall be divided by the generator polynomial $G(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$. The coefficient string of the remainder polynomial shall be appended to the data bit string to form the (15,5) BCH code string. Finally, masking shall be applied by XORing the bit string with **101010000010010** (for QR Code symbols) or **100010001000101** (for Micro QR Code symbols) to ensure that the format information bit pattern is not all zeroes for any combination of data mask pattern and Error Correction Level.

EXAMPLE

Error Correction level M; data mask pattern 101

Binary string:	**00101**
Polynomial:	$\mathbf{x^2 + 1}$
Raise power to the (15 - 5) th:	$\mathbf{x^{12} + x^{10}}$
Divide by G(x):	$\mathbf{= (x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1)x^2 + (x^7 + x^6 + x^4 + x^3 + x^2)}$

Add coefficient string of above remainder polynomial to format information data string:

00100 + 0011011100 ′ 001010011011100

XOR with mask	**101010000010010**
Result:	**100000011001110**

Place these bits in the format information areas as described in 7.9.

C.3 Error correction decoding steps

Release the masking of the format information modules by XORing the bit sequence with the mask pattern **101010000010010** (for QR Code symbols) or **100010001000101** (for Micro QR Code symbols).

The Hamming distance of the error correction code used in the format information is 7, which enables up to 3 bit errors to be corrected. There are 32 valid bit sequences for the format information, so decoding by using Table C.1 as a look-up table is efficient. Bit sequences read from the format information area of the symbol are compared with the 32 valid format information bit strings in Table C.1 on a bit by bit basis. The bit string from Table C.1 closest to the bit string read from the symbol is taken, provided the strings differ by 3 bits or less.

Example (for QR Code symbol)

Bit string read from format information area: **000011101001001**

Closest bit string from table: **000111101011001**

Since only 2 bits differ between the two bit strings, the comparison is successful, so the symbol format is confirmed as utilising error correction level M with masking pattern 011.

Table C.1 — Valid format information bit sequences

Sequence before masking		Sequence after masking (QR Code symbols)		Sequence after masking (Micro QR Code symbols)	
Data bits	Error correction bits	binary	hex	binary	hex
00000	0000000000	101010000010010	5412	100010001000101	4445
00001	0100110111	101000100100101	5125	100000101110010	4172
00010	1001101110	101111001111100	5E7C	100111000101011	4E2B
00011	1101011001	101101101001011	5B4B	100101100011100	4B1C
00100	0111101011	100010111111001	45F9	101010110101110	55AE
00101	0011011100	100000011001110	40CE	101000010011001	5099
00110	1110000101	100111110010111	4F97	101111111000000	5FC0
00111	1010110010	100101010100000	4AA0	101101011110111	5AF7
01000	1111010110	111011111000100	77C4	110011110010011	6793
01001	1011100001	111001011110011	72F3	110001010100100	62A4
01010	0110111000	111110110101010	7DAA	110110111111101	6DFD
01011	0010001111	111100010011101	789D	110100011001010	68CA
01100	1000111101	110011000101111	662F	111011001111000	7678
01101	1100001010	110001100011000	6318	111001101001111	734F
01110	0001010011	110110001000001	6C41	111110000010110	7C16
01111	0101100100	110100101110110	6976	111100100100001	7921
10000	1010011011	001011010001001	1689	000011011011110	06DE
10001	1110101100	001001110111110	13BE	000001111101001	03E9
10010	0011110101	001110011100111	1CE7	000110010110000	0CB0
10011	0111000010	001100111010000	19D0	000100110000111	0987
10100	1101110000	000011101100010	0762	001011100110101	1735
10101	1001000111	000001001010101	0255	001001000000010	1202
10110	0100011110	000110100001100	0D0C	001110101011011	1D5B
10111	0000101001	000100000111011	083B	001100001101100	186C
11000	0101001101	011010101011111	355F	010010100001000	2508
11001	0001111010	011000001101000	3068	010000000111111	203F
11010	1100100011	011111100110001	3F31	010111101100110	2F66
11011	1000010100	011101000000110	3A06	010101001010001	2A51
11100	0010100110	010010010110100	24B4	011010011100011	34E3
11101	0110010001	010000110000011	2183	011000111010100	31D4
11110	1011001000	010111011011010	2EDA	011111010001101	3E8D
11111	1111111111	010101111101101	2BED	011101110111010	3BBA

Annex D
(normative)

Version information

D.1 General

The version information consists of an 18-bit sequence comprising 6 data bits and 12 Golay error correction bits. This Annex describes the calculation of the error correction bits and the error correction decoding process.

D.2 Error correction bit calculation

The (18,6) Golay code shall be used for error correction. The polynomial whose coefficient is the data bit string shall be divided by the generator polynomial G(x) = $x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1$. The coefficient string of the remainder polynomial shall be appended to the data bit string to form the (18,6) Golay code string.

EXAMPLE

Version:	**7**
Binary string:	**000111**
Polynomial:	$\mathbf{x^2 + x + 1}$
Raise power to the (18 - 6) th:	$\mathbf{x^{14} + x^{13} + x^{12}}$
Divide by G(x):	$\mathbf{= (x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1)x^2 + (x^{11} + x^{10} + x^7 + x^4 + x^2)}$

Add coefficient string of above remainder polynomial to version information data string:

$$\mathbf{000111 + 110010010100 \rightarrow 000111110010010100}$$

Place these bits in the version information areas as described in 7.10.

Table D.1 below shows the full version information bit stream for each version.

D.3 Error correction decoding steps

The Hamming distance of the error correction code used in the version information is 8, which enables up to 3 bit errors to be corrected. There are 34 valid bit sequences for the version information, so decoding by using Table D.1 as a look-up table is efficient. Bit sequences read from the version information area of the symbol are compared with the 34 valid version information bit strings in Table D.1 on a bit by bit basis. The bit string from Table D.1 closest to the bit string read from the symbol is taken, provided the strings differ by 3 bits or less after the comparison.

EXAMPLE

Bit string read from version information area:	**000111110<u>1</u>10010100**
Closest bit string from table:	**000111110<u>0</u>10010100**

Since only 1 bit differs between the two bit strings, the comparison is successful, so the symbol version is confirmed as 7.

Table D.1 — Version information bit stream for each version

Version	Version information bit stream	Hex equivalent
7	00 0111 1100 1001 0100	07C94
8	00 1000 0101 1011 1100	085BC
9	00 1001 1010 1001 1001	09A99
10	00 1010 0100 1101 0011	0A4D3
11	00 1011 1011 1111 0110	0BBF6
12	00 1100 0111 0110 0010	0C762
13	00 1101 1000 0100 0111	0D847
14	00 1110 0110 0000 1101	0E60D
15	00 1111 1001 0010 1000	0F928
16	01 0000 1011 0111 1000	10B78
17	01 0001 0100 0101 1101	1145D
18	01 0010 1010 0001 0111	12A17
19	01 0011 0101 0011 0010	13532
20	01 0100 1001 1010 0110	149A6
21	01 0101 0110 1000 0011	15683
22	01 0110 1000 1100 1001	168C9
23	01 0111 0111 1110 1100	177EC
24	01 1000 1110 1100 0100	18EC4
25	01 1001 0001 1110 0001	191E1
26	01 1010 1111 1010 1011	1AFAB
27	01 1011 0000 1000 1110	1B08E
28	01 1100 1100 0001 1010	1CC1A
29	01 1101 0011 0011 1111	1D33F
30	01 1110 1101 0111 0101	1ED75
31	01 1111 0010 0101 0000	1F250
32	10 0000 1001 1101 0101	209D5
33	10 0001 0110 1111 0000	216F0
34	10 0010 1000 1011 1010	228BA
35	10 0011 0111 1001 1111	2379F
36	10 0100 1011 0000 1011	24B0B
37	10 0101 0100 0010 1110	2542E
38	10 0110 1010 0110 0100	26A64
39	10 0111 0101 0100 0001	27541
40	10 1000 1100 0110 1001	28C69

Annex E
(normative)

Position of alignment patterns

The alignment patterns are positioned symmetrically on either side of the diagonal running from the top left corner of the symbol to the bottom right corner. They are spaced as evenly as possible between the timing pattern and the opposite side of the symbol, any uneven spacing being accommodated between the timing pattern and the first alignment pattern in the symbol interior.

Table E.1 below shows, for each version, the number of alignment patterns and the row or column coordinates of the center module of each alignment pattern.

Table E.1 — Row/column coordinates of center module of alignment patterns

Version	Number of alignment patterns	Row/Column coordinates of center module						
1	0	-						
2	1	6	18					
3	1	6	22					
4	1	6	26					
5	1	6	30					
6	1	6	34					
7	6	6	22	38				
8	6	6	24	42				
9	6	6	26	46				
10	6	6	28	50				
11	6	6	30	54				
12	6	6	32	58				
13	6	6	34	62				
14	13	6	26	46	66			
15	13	6	26	48	70			
16	13	6	26	50	74			
17	13	6	30	54	78			
18	13	6	30	56	82			
19	13	6	30	58	86			
20	13	6	34	62	90			
21	22	6	28	50	72	94		
22	22	6	26	50	74	98		
23	22	6	30	54	78	102		
24	22	6	28	54	80	106		
25	22	6	32	58	84	110		
26	22	6	30	58	86	114		
27	22	6	34	62	90	118		

Table E.1 *(continued)*

Version	Number of alignment patterns	Row/Column coordinates of center module						
1	0	-						
28	33	6	26	50	74	98	122	
29	33	6	30	54	78	102	126	
30	33	6	26	52	78	104	130	
31	33	6	30	56	82	108	134	
32	33	6	34	60	86	112	138	
33	33	6	30	58	86	114	142	
34	33	6	34	62	90	118	146	
35	46	6	30	54	78	102	126	150
36	46	6	24	50	76	102	128	154
37	46	6	28	54	80	106	132	158
38	46	6	32	58	84	110	136	162
39	46	6	26	54	82	110	138	166
40	46	6	30	58	86	114	142	170

For example, in a Version 7 symbol the table indicates values 6, 22 and 38. The alignment patterns, therefore, are to be centered on (row, column) positions (6,22), (22,6), (22,22), (22,38), (38,22), (38,38). Note that the coordinates (6,6), (6,38), (38,6) are occupied by finder patterns and are not therefore used for alignment patterns.

Annex F
(normative)

Symbology Identifier

The Symbology Identifier assigned to QR Code in ISO/IEC 15424, which should be added as a preamble to the decoded data by a suitably programmed decoder is:

]Qm

where:

] is the Symbology Identifier flag (ASCII value 93)

Q is the code character for the QR Code symbology

m is the modifier character with one of the values defined in Table F.1.

In the case of Micro QR Code symbols, the value of m shall always be 1.

Table F.1 — Symbology Identifier options and modifier values

Modifier value	Option
0	QR Code Model 1 symbol (in accordance with AIM ITS 97-001)
1	QR Code symbol, ECI protocol not implemented
2	QR Code symbol, ECI protocol implemented
3	QR Code symbol, ECI protocol not implemented, FNC1 implied in first position
4	QR Code symbol, ECI protocol implemented, FNC1 implied in first position
5	QR Code symbol, ECI protocol not implemented, FNC1 implied in second position
6	QR Code symbol, ECI protocol implemented, FNC1 implied in second position

The permissible values of m are: 0, 1, 2, 3, 4, 5, 6.

Annex G
(normative)

QR Code print quality – symbology-specific aspects

G.1 General

Because of differences in symbology structures and reference decode algorithms, the effect of certain parameters on a symbol's reading performance may vary. ISO/IEC 15415 provides for symbology specifications to define the grading of certain symbology-specific attributes. This Annex therefore defines the method of grading Fixed Pattern Damage and additional parameters (format information and version information) to be used in the application of ISO/IEC 15415 to QR Code.

G.2 Fixed Pattern damage

G.2.1 Features to be assessed

G.2.1.1 QR Code symbols

The features to be assessed are:

— Three corner segments, each including:

  — the 7 x 7 finder pattern,

  — the 1X wide separators surrounding the two inner sides of the finder pattern,

  — part of the Quiet Zone of a minimum of four modules width (or more if specified by the application) extending for a length of 15 modules along the two outer sides of the finder pattern.

— The two timing patterns of alternating dark and light modules linking the inner corners of the finder patterns.

— The 5 x 5 alignment patterns (where present, in Model 2 symbols of Version 2 or larger).

The features listed above shall be assessed as six segments, viz.:

— the three corner segments (finder patterns with their associated separators and part of the quiet zone) (Segments A1, A2 and A3 respectively),

— the two timing patterns (Segments B1 and B2 respectively),

— the single segment containing all the alignment patterns (Segment C).

Where a timing pattern crosses an alignment pattern the five modules that coincide with the alignment pattern are assessed both as part of the timing pattern and of the alignment pattern.

In a version 7 symbol (45 x 45 modules) for example, each Segment A occupies 168 modules; each Segment B is 29 modules long, and Segment C occupies a total of 150 modules (i.e. 6 x 25).

These segments, in the case of a Version 7 symbol, are illustrated in Figure G.1 below. A1, A2 and A3 indicate the three corner segments; B1 and B2 indicate the two timing pattern segments, and C indicates the single Segment C (comprising the 6 alignment patterns).

NOTE For QR Code symbol its width of Quiet Zone shall be 4X. Figure G.1 shows segments that shall be checked at fixed pattern print quality assessment. Remaining regions of quiet zones are not checked.

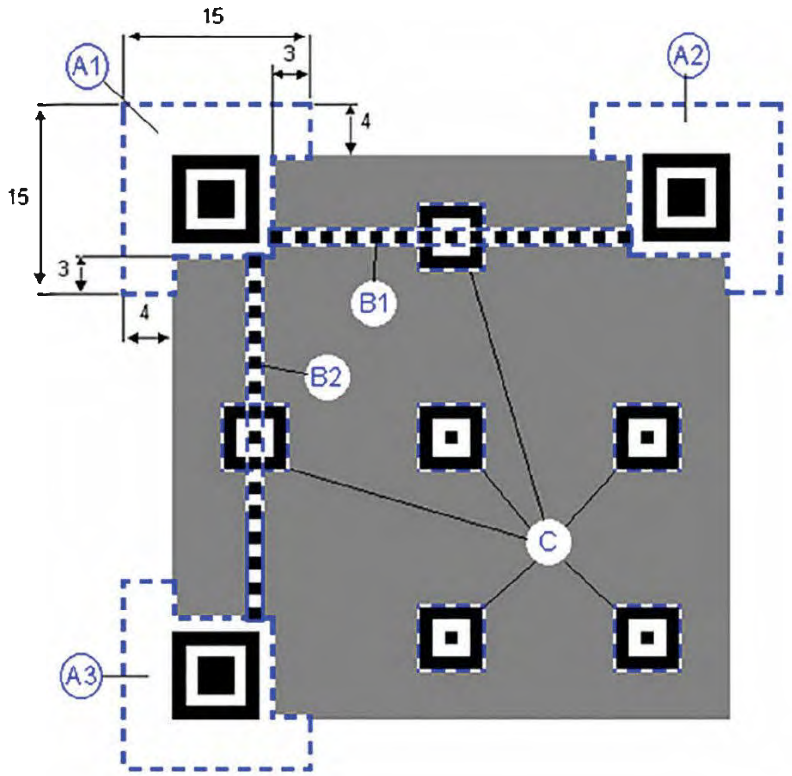


Figure G.1 — QR Code fixed pattern segments

G.2.1.2 Micro QR Code symbols

The features to be assessed are:

— The corner segment, including:

— the finder pattern,

— the 1X wide separators adjoining the two inner sides of the finder pattern,

— part of the Quiet Zone of a minimum of two modules width (or more if specified by the application) extending for a length of 11 modules along the two outer sides of the finder pattern.

— The two timing patterns of alternating dark and light modules running along the top and left side of the symbol from the finder pattern.

The features listed above shall be assessed as three segments, viz.:

— the corner segment (finder pattern with its associated separators and part of the quiet zone) (Segment A), which occupies 104 modules,

— the two timing patterns (Segments B1 and B2 respectively).

In a version M4 symbol (17 x 17 modules) for example, each Segment B is 9 modules long.

These segments, in the case of a Version M4 symbol, are illustrated in Figure G.2 below. A indicates the corner segment; and B1 and B2 indicate the two timing pattern segments.

NOTE For Micro QR Code symbol its width of Quiet Zone shall be 2X. Figure G.2 shows segment that shall be checked at fixed pattern print quality assessment. Remaining regions of quiet zones are not checked.

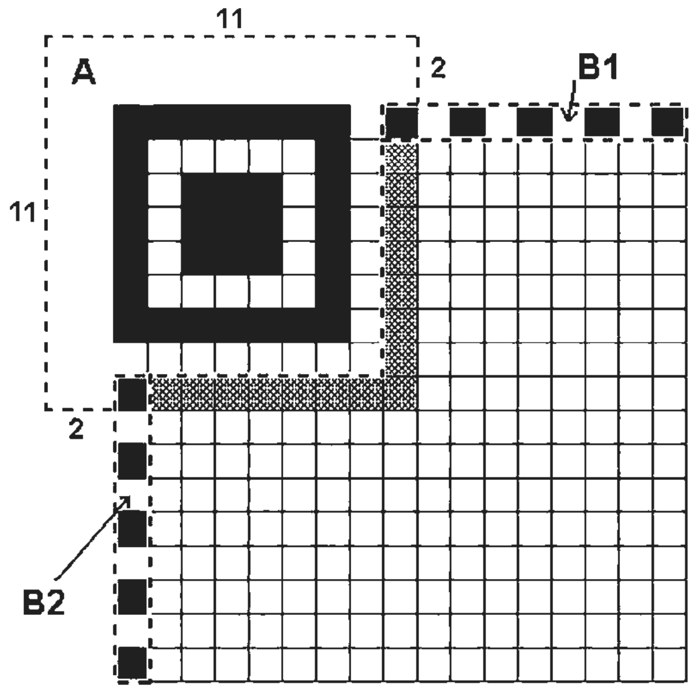


Figure G.2 — Micro QR Code fixed pattern segments

G.2.2 Fixed Pattern Damage grading

Damage to each segment shall be graded based on the modulation of the individual modules that compose it.

The procedure described below shall be applied to each segment in turn

a) From the reference grey-scale image of the symbol, find the modulation grade for each module based on the values in ISO/IEC 15415. Since the intended light or dark nature of the module is known, any module intended to be dark but the reflectance of which is above the global threshold, and any module intended to be light but the reflectance of which is below the global threshold shall be given modulation grade 0.

b) For each modulation grade level, assume that all modules not achieving that grade or a higher grade are module errors, and derive a notional damage grade based on the grade thresholds shown in Table G.1. Take the lower of the modulation grade level and the notional damage grade. The notional damage grade is determined as follows:

   1) For each of Segments A1, A2, and A3, or Segment A in Micro QR Code symbols, count the number of module errors.

2) For segments B1 and B2, count the number of module errors. Express this number as a percentage of the total number of modules in the segment.

3) For segments B1 and B2, taking groups of five adjacent modules and progressing along the segment in steps of one module, verify that in any group of five adjacent modules no more than two are damaged; if this test fails, the grade for the segment shall be 0. This test does not apply to Micro QR Code.

4) For Segment C (in QR Code symbols only), count the number of alignment patterns containing a module error. Express this number as a percentage of the number of alignment patterns in the symbol.

5) Assign a notional damage grade to each segment based on the grade thresholds shown in Table G.1.

c) The Fixed Pattern Damage grade for the segment shall be the highest resulting grade for all modulation grade levels.

The Fixed Pattern Damage grade for the symbol shall be the lowest of the segment grades.

Table G.1 — Grade thresholds for QR Code Fixed Pattern Damage

Segments A1, A2 and A3 (QR Code); Segment A (Micro QR Code)	Segments B1 and B2 (QR Code)	Segments B1 and B2 (Micro QR Code)	Segment C (QR Code)	Grade
Number of module errors	Percentage of total modules with module errors	Percentage of total modules with module errors	Percentage of alignment patterns with module errors	
0	0%	0%	0%	4
1	≤ 7%		≤ 10%	3
2	≤ 11%	≤30%	≤ 20%	2
3	≤ 14%		≤ 30%	1
≥ 4	> 14%	>30%	> 30%	0

G.3 Grading of additional parameters

G.3.1 General

QR Code symbols contain a duplicated set of modules representing information that defines the format of the symbol, and symbols of Version 7 to 40 also contain a duplicated set of modules representing information that defines the symbol size. Micro QR Code symbols contain a single set of modules representing information that defines the format of the symbol. This data requires to be reliably detected at an early stage of the decoding procedure, and if it cannot be decoded, the remainder of the symbol cannot be decoded. For this reason the format information and version information module blocks are graded separately (in a similar way to Fixed Pattern Damage), and their grades are included in the overall symbol grade determination.

G.3.2 Grading of format information

For each block of format information, determine a grade for the block according to the following method.

a) From the reference grey-scale image of the symbol, find the modulation grade for each module based on the values in ISO/IEC 15415. Since the intended light or dark nature of the module is known after decode, any module intended to be dark but the reflectance of which is above the global threshold, and any module intended to be light but the reflectance of which is below the global threshold shall

be given modulation grade 0. If the format information in the block cannot be decoded, the grade for the block shall be 0.

b) For each modulation grade level:

1) Assume that all modules not achieving that modulation grade or a higher grade are module errors, and derive a notional grade based on Table G.2:

Table G.2 — Format information notional grading

Number of module errors	Grade
0	4
1	3
2	2
3	1
≥ 4	0

2) Select the lower of the MOD grade and the notional grade at each level as the grade for that level, as illustrated in Table G.3.

3) The grade for the block shall be the highest resulting grade, as illustrated in Table G.3.

Table G.3 — Example of grading of format information block

Modulation grade	Notional grade	Lower of grades
4	2	2
3	2	2
2	3	2
1	3	1
0	4	0
	Selected (highest) Grade->	2

c) The format information grade shall be:

1) For QR Code symbols, the average of the grades of the two format information blocks, rounded up if necessary to the next integer.

2) For Micro QR Code symbols, the grade determined in step 2 c).

G.3.3 Grading of version information (QR Code symbols)

For each block of version information, determine a grade for the block according to the following method.

a) Find the modulation grade for each module based on the values in ISO/IEC 15415. Since the intended light or dark nature of the module is known after decode, any module intended to be dark but the reflectance of which is above the global threshold, and any module intended to be light but the reflectance of which is below the global threshold shall be given modulation grade 0. If the version information in the block cannot be decoded, the grade for the block shall be 0.

b) For each modulation grade level:

1) Assume that all modules not achieving that modulation grade or a higher grade are module errors, and derive a notional grade based on Table G.4:

Table G.4 — Version information notional grading

Number of module errors	Grade
0	4
1	3
2	2
3	1
≥4	0

2) Select the lower of the MOD grade and the notional grade at each level as the grade for that level, as illustrated in Table G.5.

3) The grade for the block shall be the highest resulting grade, as illustrated in Table G.5.

Table G.5 — Example of grading of version information block

Modulation grade	Notional grade	Lower of grades
4	2	2
3	2	2
2	3	2
1	3	1
0	4	0
	Selected (highest) Grade->	2

c) The version information grade shall be the average of the grades of the two version information blocks, rounded up if necessary to the next integer.

G.4 Scan grade

The scan grade shall be the lowest of the grades for the standard parameters evaluated according to ISO/IEC 15415 together with the grades for Fixed Pattern Damage, format information and (where applicable) version information evaluated in accordance with this Annex.

Annex H
(informative)

JIS8 and Shift JIS character sets

Table H.1 — 8-bit character set for JIS X 0201 (JIS8)

Char.	Hex	Char.	Hex	Char.	Hex	Char.	Hex	Char.	Hex	Char.	Hex	Char.	Hex	Char.	Hex
NUL	00	SP	20	@	40	`	60		80		A0	ﾀ	C0		E0
SOH	01	!	21	A	41	a	61		81	｡	A1	ﾁ	C1		E1
STX	02	"	22	B	42	b	62		82	｢	A2	ﾂ	C2		E2
ETX	03	#	23	C	43	c	63		83	｣	A3	ﾃ	C3		E3
EOT	04	$	24	D	44	d	64		84	､	A4	ﾄ	C4		E4
ENQ	05	%	25	E	45	e	65		85	･	A5	ﾅ	C5		E5
ACK	06	&	26	F	46	f	66		86	ｦ	A6	ﾆ	C6		E6
BEL	07	'	27	G	47	g	67		87	ｧ	A7	ﾇ	C7		E7
BS	08	(	28	H	48	h	68		88	ｨ	A8	ﾈ	C8		E8
HT	09	)	29	I	49	i	69		89	ｩ	A9	ﾉ	C9		E9
LF	0A	*	2A	J	4A	j	6A		8A	ｪ	AA	ﾊ	CA		EA
VT	0B	+	2B	K	4B	k	6B		8B	ｫ	AB	ﾋ	CB		EB
FF	0C	,	2C	L	4C	l	6C		8C	ｬ	AC	ﾌ	CC		EC
CR	0D	-	2D	M	4D	m	6D		8D	ｭ	AD	ﾍ	CD		ED
SO	0E	.	2E	N	4E	n	6E		8E	ｮ	AE	ﾎ	CE		EE
SI	0F	/	2F	O	4F	o	6F		8F	ｯ	AF	ﾏ	CF		EF
DLE	10	0	30	P	50	p	70		90	ｰ	B0	ﾐ	D0		F0
DC1	11	1	31	Q	51	q	71		91	ｱ	B1	ﾑ	D1		F1
DC2	12	2	32	R	52	r	72		92	ｲ	B2	ﾒ	D2		F2
DC3	13	3	33	S	53	s	73		93	ｳ	B3	ﾓ	D3		F3
DC4	14	4	34	T	54	t	74		94	ｴ	B4	ﾔ	D4		F4
NAK	15	5	35	U	55	u	75		95	ｵ	B5	ﾕ	D5		F5
SYN	16	6	36	V	56	v	76		96	ｶ	B6	ﾖ	D6		F6
ETB	17	7	37	W	57	w	77		97	ｷ	B7	ﾗ	D7		F7
CAN	18	8	38	X	58	x	78		98	ｸ	B8	ﾘ	D8		F8
EM	19	9	39	Y	59	y	79		99	ｹ	B9	ﾙ	D9		F9
SUB	1A	:	3A	Z	5A	z	7A		9A	ｺ	BA	ﾚ	DA		FA
ESC	1B	;	3B	[	5B	{	7B		9B	ｻ	BB	ﾛ	DB		FB
FS	1C	<	3C	¥	5C	|	7C		9C	ｼ	BC	ﾜ	DC		FC
GS	1D	=	3D	]	5D	}	7D		9D	ｽ	BD	ﾝ	DD		FD
RS	1E	>	3E	^	5E	‾	7E		9E	ｾ	BE	ﾞ	DE		FE
US	1F	?	3F	_	5F	DEL	7F		9F	ｿ	BF	ﾟ	DF		FF

Figure H.1 below shows the areas of the 256 x 256 code plane occupied by Shift JIS double byte characters.

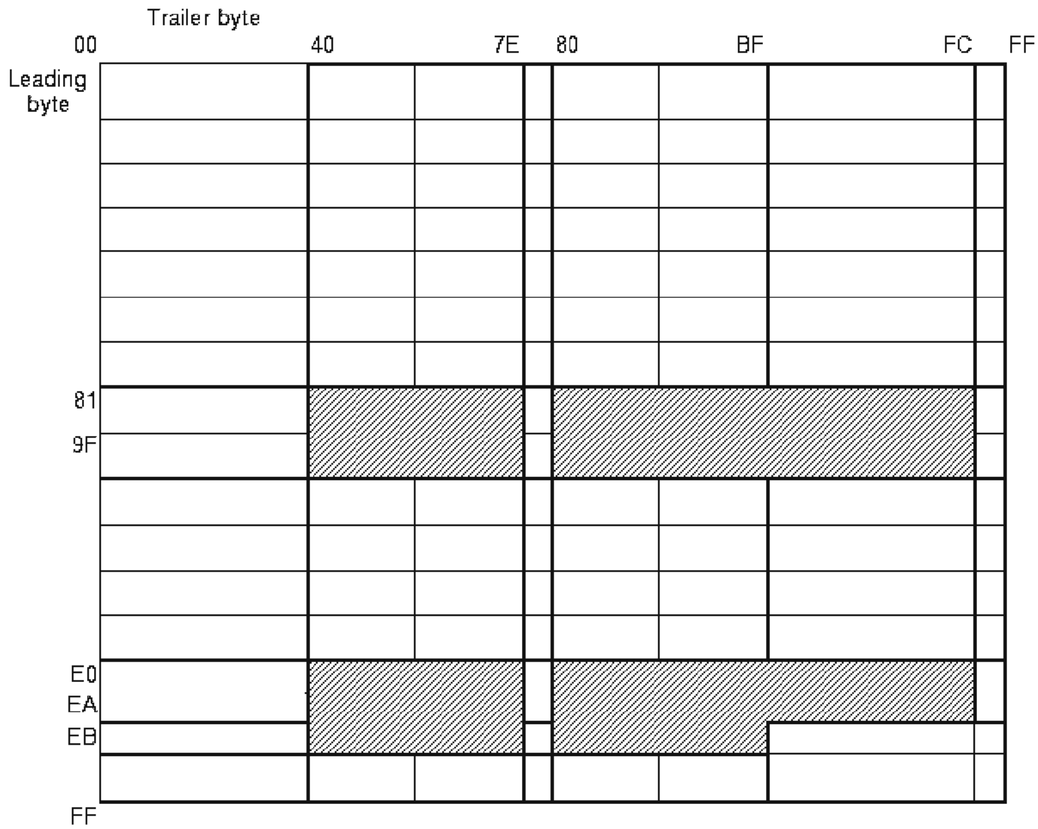


Figure H.1 — Shift JIS character values

According to JIS X 0208:1997, Annex 1, leading and trailing bytes within the ranges shown shaded are assigned to Shift JIS Kanji characters. Any pairs of bytes within these ranges may be encoded using the Kanji mode compaction scheme.

Annex I
(informative)

Symbol encoding examples

I.1 General

This Annex describes the encoding of the data string **01234567** into both a QR Code symbol and a Micro QR Code symbol.

I.2 Encoding a QR Code symbol

The data string is to be encoded into a version 1-M symbol, using the Numeric mode in accordance with 7.4.3.

Step 1: Data Encoding

— Divide into groups of three digits and convert each group to its 10 or 7-bit binary equivalent:

 — 012 → 0000001100

 — 345 → 0101011001

 — 67 → 1000011

— Convert character count indicator to binary (10 bits for version 1-M)

Character count indicator (8) = **0000001000**

— Connect mode indicator for Numeric mode (**0001**), character count indicator, binary data, and Terminator (**0000**)

0001 0000001000 0000001100 0101011001 1000011 0000

— Divide into 8-bit codewords, adding padding bits (shown underlined for illustration) as needed

00010000 00100000 00001100 01010110 01100001 10000000

— Add Pad codewords to fill data codeword capacity of symbol (for version 1-M, 16 data codewords, therefore 10 Pad codewords required (shown underlined for illustration)), giving the result:

00010000 00100000 00001100 01010110 01100001 10000000 11101100 00010001 11101100 00010001 11101100 00010001 11101100 00010001 11101100 00010001

Step 2: Error Correction Codeword generation

Using the Reed-Solomon algorithm to generate the required number of error correction codewords (for a Version 1-M symbol, 10 are needed), these (shown underlined for illustration) should be added to the bit stream, resulting in:

00010000 00100000 00001100 01010110 01100001 10000000 11101100 00010001 11101100 00010001 11101100 00010001 11101100 00010001 11101100 00010001 10100101 00100100 11010100 11000001 11101101 00110110 11000111 10000111 00101100 01010101

Step 3: Module placement in matrix

As there is only a single error correction block in a version 1-M symbol, no interleaving is required in this instance. The finder patterns, separators and timing patterns are placed in a blank 21 × 21 matrix and the module positions for the format information are left temporarily blank. The codewords from Step 2 are placed in the matrix in accordance with 7.7.3, which results in the arrangement shown in Figure I.1.

Figure I.1 — Data modules placed in symbol prior to data masking

Step 4: Data masking pattern selection

Apply the data masking patterns defined in 7.8.2 in turn and evaluate the results in accordance with 7.8.3. The data masking pattern selected is referenced **010**.

Step 5: Format information

The error correction level is M and the data masking pattern is 011. Therefore, from 7.9.1 the data bits of the format information are **00 010**.

The BCH error correction calculation gives **1001101110** as the bit sequence to be added to the data, giving:

000101001101110 as the unmasked format information.

XOR this bit stream with the mask **101010000010010**:

000101001101110 (raw bit stream)

101010000010010 (mask)

101111001111100 (format information to be placed in symbol)

Step 6: Final symbol construction

Apply the selected data masking pattern to the encoding region of the symbol as described in 7.8, and add format information modules in positions reserved in step 3. The final symbol is shown in Figure I.2.

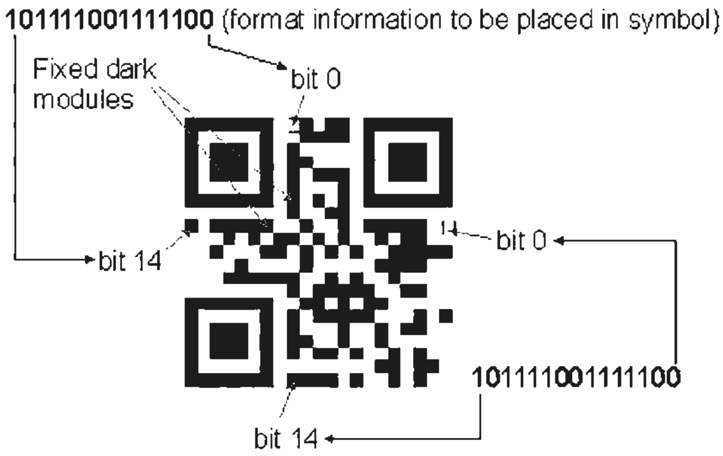


Figure I.2 — Final version 1-M symbol encoding 01234567

I.3 Encoding a Micro QR Code symbol

The data string **01234567** is to be encoded into a Version M2 symbol with EC level L, using the Numeric mode in accordance with 7.4.3.

Step 1: Data Encoding

— Divide into groups of three digits and convert each group to its 10 or 7-bit binary equivalent:

— 012 → 0000001100

— 345 → 0101011001

— 67 → 1000011

— Mode indicator for Numeric mode in Version M2 is **0**

— Character count is 8; convert to binary (4 bits for Version M2-L):

Character count indicator (8) = **1000**

— Terminator for Version M2 is 5 zero bits, **00000**

— Connect mode indicator for Numeric mode (**0**), character count indicator (**1000**), binary data, and Terminator (**00000**)

0 1000 0000001100 0101011001 1000011 00000

— Divide into 8-bit codewords, adding 3 padding bits (shown underlined for illustration) since final codeword contained only 5 bits

01000000 00011000 10101100 11000011 00000<u>000</u>

— No Pad codewords are required to fill data codeword capacity of symbol (for version M2-L, 5 data codewords).

Step 2: Error Correction Codeword generation

Using the Reed-Solomon algorithm to generate the required number of error correction codewords (for a Version M2-L symbol, 5 are needed), these (shown underlined for illustration) should be added to the bit stream, resulting in:

01000000 00011000 10101100 11000011 00000000 10000110 00001101 00100010 10101110 00110000

Step 3: Module placement in matrix

The finder pattern and timing patterns are placed in a blank 13 × 13 matrix and the module positions for the format information are left temporarily blank. The codewords from Step 2 are placed in the matrix in accordance with 7.7.3. Figure I.3 shows the module arrangement.

Figure I.3 — Data modules placed in symbol prior to data masking

Step 4: Data masking pattern selection

Apply the data masking patterns defined in 7.8.2 in turn and evaluate the results in accordance with 7.8.3. The data masking pattern selected is referenced **01**. Apply the selected data masking pattern to the encoding region of the matrix, as described in 7.8.

Step 5: Format information

The symbol number for an M2-L symbol is 1, which is represented in binary form as **001**, and the data masking pattern is **01**. Therefore, the data bits of the format information are **001 01**.

The BCH error correction calculation gives **0011011100** as the bit sequence to be added to the data, giving:

001010011011100 as the unmasked format information.

XOR this bit stream with the mask **100010001000101**:

— **001010011011100** (raw bit stream)

— **100010001000101** (mask)

— **101000010011001** (format information to be placed in symbol)

Step 6: Final symbol construction

Add format information modules in positions reserved in step 3. The final symbol is shown in Figure I.4.

Figure I.4 — Final version M2-L symbol encoding 01234567

Annex J
(informative)

Optimisation of bit stream length

J.1 General

As described in this standard, QR Code offers various modes of encoding each of which differs in the number of bits it requires to represent a given data string. Since there is an overlap between the character sets of each mode - for example, numeric data may be encoded in Numeric, Alphanumeric and Byte modes, and Latin alphanumeric data may be encoded in Alphanumeric and Byte modes - the symbol generation software may need to choose the most appropriate mode in which to encode data characters which appear in more than one mode.

A choice may also be possible between a QR Code symbol and a Micro QR Code symbol.

The choice of mode must be made initially and the mode may be changed part way through a data stream.

A number of alternative approaches may be adapted to minimize the bit stream length. The algorithm will need not only to consider the immediate sequence of characters but also look ahead to the next sequence of data in view of the overhead required for switching modes. The term "exclusive subset" is used in this Annex as a short way of referring to the set of characters within the character set of a mode which are not shared with the more restricted character set of another mode, as shown below and in Table J.1.

The numeric exclusive subset is the set of hex values 30 to 39 (digits 0 to 9).

The Alphanumeric exclusive subset is the set of hex values 20, 24, 25, 2A, 2B, 2D to 2F, 3A, and 41 to 5A, mapped as {A - Z, space, $ % * + - . / :}.

NOTE 1 This subset does not include the digits.

The Byte exclusive subset comprises hex values 00 - FF, but excludes hex values 20, 24, 25, 2A, 2B, 2D - 3A, and 41 - 5A.

NOTE 2 The excluded values are contained in the alphanumeric and numeric exclusive subsets.

Table J.1 — Exclusive subset byte values for QR Code modes

Exclusive subset	Byte values (hex)
Numeric	30 to 39
Alphanumeric	20, 24, 25, 2A, 2B, 2D to 2F, 3A, and 41 to 5A
Byte	00 to 1F, 21 to 23, 26 to 29, 2C, 3B to 40, 5B to FF (excluding reserved values 80 to 9F and E0 to FF)
Kanji	All double bytes in ranges defined in Annex H

The compaction efficiencies given in 7.4.3 to 7.4.6 need to be interpreted carefully. The best scheme for a given set of data may not be the one with the fewest bits per data character. If the highest degree of compaction is required, account has to be taken of the additional bits required to change modes (additional mode indicator and character count indicator). It should also be noted that even if the number of codewords is minimized, the codeword stream may need to be expanded to fill a symbol. This fill process is done using pad characters.

J.2 Optimisation for QR Code symbols

For QR Code symbols, the following guidelines form the basis of one possible algorithm to determine the shortest bit stream for any given input data.

Numbers of characters shown in square brackets e.g.[5,7,9] are applicable to versions 1 - 9, 10 - 26, and 27 - 40 respectively.

a) Select initial mode:

1) If initial input data is in the exclusive subset of the Byte character set, select Byte mode;

2) If initial input byte is in the Kanji leading byte exclusive subset and the next byte is in the Kanji trailing byte exclusive subset, AND the subsequent data is in the Alphanumeric or Numeric exclusive character set, select Kanji mode, ELSE if subsequent data is in the Byte exclusive character set AND the following[5,5,6] byte pairs are also in the Kanji exclusive subsets, select Byte mode;

3) If initial input data is in the exclusive subset of the Alphanumeric character set AND if there are less than[6-8] characters followed by data from the remainder of the Byte character set, THEN select the Byte mode ELSE select Alphanumeric mode;

4) If initial data is numeric, AND if there are less than[4,4,5] characters followed by data from the exclusive subset of the Byte character set, THEN select Byte mode ELSE IF there are less than[7-9] characters followed by data from the exclusive subset of the Alphanumeric character set THEN select Alphanumeric mode ELSE select Numeric mode.

b) While in Byte mode:

1) If a sequence of at least[9,12,13] byte pairs from the Kanji set occurs before more data from the exclusive subset of the Byte character set, switch to Kanji mode;

2) If a sequence of at least[11,15,16] character from the exclusive subset of the Alphanumeric character set occurs before more data from the exclusive subset of the Byte character set, switch to Alphanumeric mode;

3) If a sequence of at least[6,8,9] Numeric characters occurs before more data from the exclusive subset of the Byte character set, switch to Numeric mode;

4) If a sequence of at least[6-8] Numeric characters occurs before more data from the exclusive subset of the Alphanumeric character set, switch to Numeric mode.

c) While in Alphanumeric mode:

1) If one or more Kanji characters occurs, switch to Kanji mode;

2) If one or more characters from the exclusive subset of the Byte character set occurs, switch to Byte mode;

3) If a sequence of at least[13,15,17] Numeric characters occurs before more data from the exclusive subset of the Alphanumeric character set, switch to Numeric mode.

d) While in Numeric mode:

1) If one or more Kanji character occurs, switch to Kanji mode;

2) If one or more characters from the exclusive subset of the Byte character set occurs, switch to Byte mode;

3) If one or more characters from the exclusive subset of the Alphanumeric character set occurs, switch to Alphanumeric mode.

J.3 Optimisation for Micro QR Code symbols

J.3.1 Optimisation principles

Assuming that the data to be encoded is in the exclusive subsets of not more than two modes, and that all the data in each subset is grouped together (e.g. "123abcdef"), an algorithm to determine the shortest bit stream for Micro QR Code data can be derived from Table J.2. These principles can be extended to cater for more than two modes, although care must be taken that the resulting bit stream will fit one of the available symbols.

Because the lower modes use fewer bits per character than the higher modes, there is a point at which the extra overhead of the additional mode indicator and character count indicator for a change of mode is offset by the greater encoding density of the lower mode. Table J.2 shows the minimum number of consecutive characters in a lower mode for which a shorter total bit stream is achieved by changing modes. For fewer characters, encoding all the data in the higher mode will give a shorter bit stream.

Table J.2 — Minimum characters in lower mode for minimising bit stream length by changing modes

Mode combination	M2 symbols	M3 symbols	M4 symbols
Numeric + Alphanumeric	3 numeric	4 numeric	5 numeric
Numeric + 8-bit byte	n/a	2 numeric	3 numeric
Alphanumeric + Byte	n/a	3 alphanumeric	4 alphanumeric

J.3.2 Capacity of Micro QR Code symbols

Based on the principles of the above table, and the capacities of the various symbol versions, Figures J.1 to J.6 below show, for each combination of modes, the options available for encoding given amounts of data in combinations of modes.

The column and row headings identify the number of characters in each mode. The figures show the symbol versions and error correction levels, omitting the initial M; thus, for example, 4Q refers to a version M4 symbol with error correction level Q. For any given combination of characters and modes, the available symbol versions are those at the appropriate row and column intersection and those shown to the right of or below that intersection.

For example, if the data string was "123456ABCDEFGH", consisting of six numeric characters and eight from the alphanumeric character set, Figure J.1 shows that the data would fit into a version M3-L symbol (total of 77 bits including mode indicators and character count indicators), or a version M4-M symbol or a version M4-L symbol (81 bits for either). The options may be narrowed down either by the space available or the required level of error correction.

	Alphanumeric																					
Num.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0						2M	2L					3M		4Q	3L				4M			4L
1	1				2M	2L					3M		4Q	3L				4M			4L	
2	1			2M	2L					3M		4Q	3L				4M			4L		
3	1		2M	2L					3M		4Q	3L				4M			4L			
4	1	2M		2L				3M		4Q	3L				4M			4L				
5	1	2M	2L				3M		4Q	3L					4M			4L				
6	2M		2L				3M	4Q		3L				4M			4L					
7	2M	2L				3M		4Q	3L				4M			4L						
8	2M					3M	4Q		3L				4M			4L						
9	2L				3M		4Q	3L				4M			4L							
10	2L			3M		4Q	3L					4M			4L							
11				3M	4Q	3L					4M			4L								
12			3M		4Q	3L					4M		4L									
13			3M	4Q	3L					4M			4L									
14		3M	4Q		3L				4M			4L										
15	3M		4Q	3L					4M			4L										
16	3M	4Q		3L				4M			4L											
17	3M	4Q	3L				4M			4L												
18	3M/4Q		3L				4M			4L												
19	4Q	3L				4M			4L													
20	3L/4Q					4M			4L													
21	3L/4Q				4M			4L														
22	3L			4M			4L															
23	3L			4M			4L															
24			4M			4L																
25			4M		4L																	
26		4M			4L																	
27	4M			4L																		
28	4M			4L																		
29	4M		4L																			
30	4M		4L																			
31		4L																				
32	4L																					
33	4L																					
34	4L																					
35	4L																					

Figure J.1 — Micro QR Code symbol capacities - numeric and alphanumeric data

Num.	Byte															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0								3M		3L/4Q				4M		4L
1							3M		3L/4Q				4M		4L	
2							3M	4Q	3L			4M		4L		
3						3M	4Q	3L			4M		4L			
4						3M	4Q	3L			4M		4L			
5	1				3M	4Q	3L			4M		4L				
6	2M				3M	4Q	3L			4M		4L				
7	2M			3M	4Q	3L			4M		4L					
8	2M			3M	4Q	3L			4M		4L					
9	2L			3M	4Q	3L			4M		4L					
10	2L		3M	4Q	3L			4M		4L						
11			3M	4Q	3L			4M		4L						
12		3M	4Q	3L			4M		4L							
13		3M	4Q	3L			4M		4L							
14		3M	4Q	3L			4M		4L							
15	3M	4Q	3L			4M		4L								
16	3M	4Q	3L			4M		4L								
17	3M/4Q	3L			4M		4L									
18	3M/4Q	3L			4M		4L									
19	3L/4Q			4M		4L										
20	3L/4Q			4M		4L										
21	3L/4Q			4M		4L										
22	3L		4M		4L											
23	3L		4M		4L											
24		4M		4L												
25		4M		4L												
26		4M		4L												
27	4M		4L													
28	4M		4L													
29	4M	4L														
30	4M	4L														
31	4L															
32	4L															
33	4L															
34	4L															
35	4L															

Figure J.2 — Micro QR Code symbol capacities - numeric and Byte data

A/num.	Byte 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0								3M		3L/4Q				4M		4L
1	2M						3M		3L/4Q				4M		4L	
2	2M					3M		3L/4Q				4M		4L		
3	2M				3M		3L/4Q				4M		4L			
4	2M				3M	4Q	3L			4M		4L				
5	2M			3M	4Q	3L			4M		4L					
6	2L		3M	4Q	3L			4M		4L						
7			3M	4Q	3L			4M		4L						
8		3M	4Q	3L			4M		4L							
9	3M	4Q	3L			4M		4L								
10	3M	4Q	3L			4M		4L								
11	3M/4Q	3L			4M		4L									
12	3L/4Q			4M		4L										
13	3L/4Q			4M		4L										
14	3L		4M		4L											
15		4M		4L												
16		4M		4L												
17	4M		4L													
18	4M	4L														
19	4L															
20	4L															
21	4L															

Figure J.3 — Micro QR Code symbol capacities - alphanumeric and Byte data

Num.	Kanji 0	1	2	3	4	5	6	7	8	9
0					3M	4Q	3L		4M	4L
1	1				3M/4Q	3L		4M	4L	
2	1			3M	4Q	3L	4M		4L	
3	1			3M	3L/4Q		4M	4L		
4	1			3M/4Q	3L		4M	4L		
5	1			3M/4Q	3L		4M	4L		
6	2M		3M	4Q	3L	4M		4L		
7	2M		3M	3L/4Q		4M	4L			
8	2M		3M/4Q	3L		4M	4L			
9	2L		3M/4Q	3L		4M	4L			
10	2L	3M	3L/4Q		4M		4L			
11		3M	3L/4Q		4M	4L				
12		3M/4Q	3L		4M	4L				
13	3M	4Q	3L		4M	4L				
14	3M	3L/4Q		4M		4L				
15	3M	3L/4Q		4M	4L					
16	3M/4Q	3L		4M	4L					
17	3M/4Q	3L		4M	4L					
18	3M/4Q		4M		4L					
19	3L/4Q		4M	4L						
20	3L/4Q		4M	4L						
21	3L/4Q		4M	4L						
22	3L	4M	4L							
23	3L	4M	4L							
24		4M	4L							
25	4M		4L							
26	4M	4L								
27	4M	4L								
28	4M	4L								
29	4M	4L								
30	4M									
31	4L									
32	4L									
33	4L									
34	4L									
35	4L									

Figure J.4 — Micro QR Code symbol capacities - numeric and Kanji data

	Kanji									
A/num.	0	1	2	3	4	5	6	7	8	9
0					3M	4Q	3L		4M	4L
1	2M			3M	4Q	3L		4M	4L	
2	2M			3M	3L/4Q		4M	4L		
3	2M			3M/4Q	3L		4M	4L		
4	2M		3M	3L/4Q		4M		4L		
5	2M		3M/4Q	3L		4M	4L			
6	2L	3M	4Q	3L	4M		4L			
7		3M	3L/4Q		4M	4L				
8		3M/4Q	3L		4M	4L				
9	3M	3L/4Q		4M	4L					
10	3M/4Q	3L		4M	4L					
11	3M/4Q		4M		4L					
12	3L/4Q		4M	4L						
13	3L/4Q	4M		4L						
14	3L	4M	4L							
15		4M	4L							
16	4M	4L								
17	4M	4L								
18	4M									
19	4L									
20	4L									
21	4L									

Figure J.5 — Micro QR Code symbol capacities - alphanumeric and Kanji data

	Kanji									
8-bit	0	1	2	3	4	5	6	7	8	9
0					3M	4Q	3L		4M	4L
1				3M	4Q	3L		4M	4L	
2				3M/4Q	3L		4M	4L		
3			3M	3L/4Q		4M	4L			
4		3M	4Q	3L		4M	4L			
5		3M/4Q	3L		4M	4L				
6	3M	3L/4Q		4M		4L				
7	4Q	3L		4M	4L					
8	3L/4Q		4M	4L						
9	3L/4Q	4M		4L						
10		4M	4L							
11	4M	4L								
12	4M	4L								
13	4M									
14	4L									
15	4L									

Figure J.6 — Micro QR Code symbol capacities - Byte and Kanji data

Annex K
(informative)

User guidelines for printing and scanning of QR Code symbols

K.1 General

Any QR Code application must be viewed as a total system solution. All the symbology encoding/decoding components (surface markers or printers, labels, readers) making up an installation need to operate together as a system. A failure in any link of the chain, or a mismatch between them, could compromise the performance of the overall system.

While compliance with the specifications is one key to assuring overall system success, other considerations come into play which may influence performance as well. The following guidelines suggest some factors to keep in mind when specifying or implementing bar or matrix code systems:

a) Select a print density which will yield tolerance values that can be achieved by the marking or printing technology being used. Ensure that the module dimension is an integer multiple of the print head pixel dimension (both parallel to and perpendicular to the print direction). Ensure also that any adjustment for print gain (or loss) is performed by changing an equal integer number of pixels from dark to light (or light to dark) on all dark-to-light boundaries of individual or groups of adjoining dark modules in order to ensure that the module center spacing remains constant, although the apparent bit-map representation of the individual dark (or light) modules is adjusted in size to suit the direction of compensation.

b) Choose a reader with a resolution suitable for the symbol density and quality produced by the marking or printing technology.

c) Ensure that the optical properties of the printed symbol are compatible with the wavelength of the scanner light source or sensor.

d) Verify symbol compliance in the final label or package configuration. Overlays, show-through and curved or irregular surfaces can all affect symbol readability.

The effects of specular reflection from glossy symbol surfaces must be considered. Scanning systems must take into account the variations in diffuse reflection between dark and light features. At some scanning angles, the specular component of the reflected light can greatly exceed the desired diffuse component, changing the scanning performance. In cases where the surface of the material or part can be altered, matt, non-glossy surfaces may help minimize specular effects. Where this option is not available, particular must be taken to ensure the illumination of the symbol to be read optimizes the desired contrast components.

K.2 User selection of error correction level

The users should define the appropriate level of error correction to suit the application requirements. As shown in Table 8, the four levels from L to H offer increasing capabilities of detecting and correcting errors, at the cost of some increase in symbol size for a given message length. For example, a Version 20-Q symbol can contain a total of 485 data codewords, but if a lower level of error correction was acceptable, the same data could also be represented in a Version 15-L symbol (exact capacity 523 data codewords).

The error correction level should be determined in relation to:

— the expected level of symbol quality: the lower the expected quality grade, the higher the level to be applied;

— the importance of a high first read rate;

— the opportunity for re-scanning in the event of a read failure;

— the space constraints which might reduce the opportunity to use a higher error correction level.

Error correction level L is appropriate for high symbol quality and/or the need for the smallest possible symbol for given data. Level M is described as "Standard" level and offers a good compromise between small size and increased reliability. Level Q is a "High reliability" level and suitable for more critical or poor print quality applications while level H offers the maximum achievable reliability.

Annex L
(informative)

Autodiscrimination

QR Code may be read by suitably programmed decoders which have been designed to autodiscriminate it from other symbologies. A properly programmed QR Code reader will not decode a symbol in another symbology as a valid QR Code symbol; however, representations of short linear symbols may be found in any matrix symbol including QR Code.

Although QR Code Model 1 symbols can be autodiscriminated from QR Code symbols by a suitable decoder, it is strongly recommended that the two types of symbol should not be mixed in an application.

The decoder's valid set of symbologies should be limited to those needed by a given application in order to maximize reading security.

Annex M
(informative)

Process control techniques

M.1 General

This Annex describes tools and procedures useful for monitoring and controlling the process of creating scannable QR Code symbols. These techniques do not constitute a print quality check of the produced symbols - the method defined in 10 and Annex G is the required method for assessing symbol quality - but they individually and collectively yield good indications of whether the symbol production process is creating workable symbols.

M.2 Symbol Contrast

Most verifiers for linear bar code symbols have either a reflectometer mode or a mode for plotting scan reflectance profiles and/or reporting Symbol Contrast, as defined in ISO/IEC 15416, from undecodable scans. Except with symbols requiring special illumination configurations, the symbol contrast readings that can be obtained using a 0,150 mm or 0,250 mm aperture at 660 nm wavelength - either the reported symbol contrast value, the maximum to minimum scan reflectance profile excursions, or the difference between maximum and minimum reflectometer readings - are found to correlate well with an image-derived symbol contrast value. In particular these reading can be used to check that symbol contrast stays well above the minimum allowed for the intended symbol quality grade.

M.3 Assessing Axial Nonuniformity

For a QR Code symbol, measure the distance from the left edge of the upper left finder pattern to the right edge of the upper right finder pattern, and the distance from the top edge of the upper left finder pattern to the bottom edge of the lower left finder pattern. For a Micro QR Code symbol, measure the distance from the left edge of the upper left finder pattern to the right edge of the rightmost module in the upper timing pattern, and the distance from the top edge of the upper left finder pattern to the bottom edge of the lowest module in the left side timing pattern. Divide each of these by the number of modules in that dimension. E.g. a version 2 symbol would have 25 as a divisor. Substitute the results for X_{AVG} and Y_{AVG} in the formula in G.2.4 and grade the result for an assessment of Axial Nonuniformity.

M.4 Visual inspection for symbol distortion and defects

Ongoing visual inspection of the Finder and timing patterns in sample symbols can monitor an important aspect of the production process.

Matrix code symbols are susceptible to errors caused by local distortions of the matrix grid. Any such distortions may show up visually as either crooked edges on the finder patterns or uneven spacings within the alternating timing patterns running between the finder patterns and aligned with the inner boundaries of these.

The finder patterns and the adjacent quiet zone areas should always be solidly dark and light. Failures in the print mechanism which may produce defects in the form of light or dark streaks through the symbol should be visibly evident where they traverse the finder pattern or the quiet zone. Such systematic failures in the print process should be corrected.

M.5 Assessing print growth

A linear bar code verifier capable of outputting direct measurements of bar and space patterns may be used for the assessment of print gain or loss in both horizontal and vertical axes, by measuring along two scan paths at right angles passing through a finder pattern and crossing the center 3 x 3 block of modules. Analysis of the output should reveal an apparent bar/space/bar/space/bar pattern; the print gain (or loss) can be assessed by comparing the five measured element widths with the ideal 1 : 1 : 3 : 1 : 1 ratio of the widths.

Annex N
(informative)

Characteristics of Model 1 symbols

N.1 Model 1 QR Code symbols

Model 1 of QR Code, as defined in AIM ITS 97-001, is the form of the symbology originally used for a number of early or closed systems applications but is not recommended for use in new or open systems applications, and is unsuitable where data volumes are likely to be high. In most respects it follows the same specification as QR Code but differs in a number of significant aspects which are summarised in this Annex.

N.1.1 Model 1 overall characteristics

a) Symbol size (not including quiet zone):

   21 × 21 modules to 73 × 73 modules (Versions 1 to 14, increasing in steps of 4 modules per side).

b) Maximum data capacity (for maximum symbol size with lowest level of error correction, Version 14-L):

   — numeric data: 1 167 characters

   — alphanumeric data: 707 characters

   — Byte data: 486 characters

   — Kanji data: 299 characters

c) Symbol structure and features compared with QR Code:

   — alignment patterns: Model 1 symbols have no alignment patterns

   — extension patterns: Model 1 symbols have extension patterns on the right-hand and lower sides

   — version information : Model 1 symbols contain no version information

   — symbol character placement : in consequence of the above, symbol character placement follows different rules.

   — Model 1 symbols do not support the ECI protocol

   — Model 1 symbols do not support mirror imaging

   — Model 1 symbols do not support reflectance reversal

d) Error correction: the error detection and correction codewords are calculated identically with QR Code, but the number and size of error correction blocks for any Version differ. Data and error correction codeword blocks are not subject to interleaving.

Figure N.1 below illustrates the structure of a Version 7 Model 1 QR Code symbol.

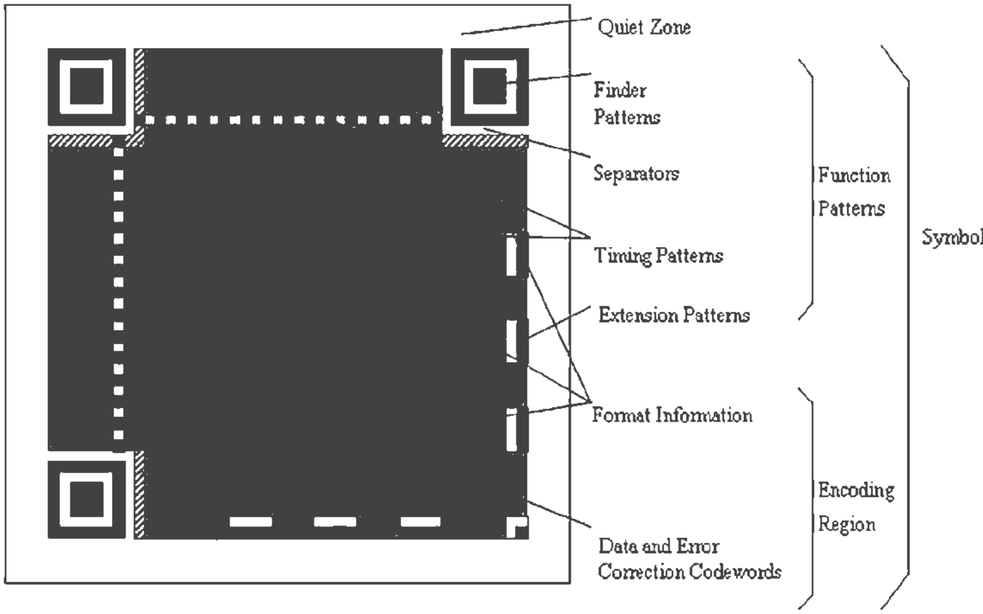


Figure N.1 — Structure of a QR Code Model 1 symbol

N.1.2 Symbol versions and sizes

There are only fourteen sizes of Model 1 symbol, from Version 1 to Version 14, the sizes of which are identical with those of Model 2 symbols with the same Version numbers, as defined in 6.3.2. Version 1 symbols therefore measure 21 × 21 modules, and Version 14 symbols 73 × 73 modules. Table N.1 shows the data capacity of all Model 1 symbols at the different error correction levels.

Table N.1 — Data capacity of all versions of Model 1 QR Code

Version	No. of Modules/side (A)	Function Patterns Modules (B)	Format Information Modules (C)	Data Modules except (C) (D=A²-B-C)	Data Capacity [codewords]* (E)
1	21	206	31	204	26
2	25	230	31	364	46
3	29	238	31	572	72
4	33	262	31	796	100
5	37	270	31	1 068	134
6	41	294	31	1 356	170
7	45	302	31	1 692	212
8	49	326	31	2 044	256
9	53	334	31	2 444	306
10	57	358	31	2 860	358
11	61	366	31	3 324	416
12	65	390	31	3 804	476

Table N.1 *(continued)*

Version	No. of Modules/side (A)	Function Patterns Modules (B)	Format Information Modules (C)	Data Modules except (C) ($D=A^2$-B-C)	Data Capacity [codewords]* (E)
13	69	398	31	4 332	542
14	73	422	31	4 876	610

NOTE The first codeword is 4 bits in length. All subsequent codewords are 8 bits in length. The first, 4-bit, data codeword is prefixed with 0000 to make its length 8 bits for generating the error correction codewords.

N.2 Detailed specifications

For complete information regarding the printing and reading of Model 1, see AIM ITS 97-001.

Bibliography

[1] ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

[2] ISO/IEC 8859-2:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 2: Latin alphabet No. 2*

[3] ISO/IEC 8859-3:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 3: Latin alphabet No. 3*

[4] ISO/IEC 8859-4:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 4: Latin alphabet No. 4*

[5] ISO/IEC 8859-5:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 5: Latin/Cyrillic alphabet*

[6] ISO/IEC 8859-6:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 6: Latin/Arabic alphabet*

[7] ISO/IEC 8859-7:2003, *Information technology — 8-bit single-byte coded graphic character sets — Part 7: Latin/Greek alphabet*

[8] ISO/IEC 8859-8:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 8: Latin/Hebrew alphabet*

[9] ISO/IEC 8859-9:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 9: Latin alphabet No. 5*

[10] ISO/IEC 8859-10:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 10: Latin alphabet No. 6*

[11] ISO/IEC 8859-11:2001, *Information technology — 8-bit single-byte coded graphic character sets — Part 11: Latin/Thai alphabet*

[12] ISO/IEC 8859-13:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 13: Latin alphabet No. 7*

[13] ISO/IEC 8859-14:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 14: Latin alphabet No. 8 (Celtic)*

[14] ISO/IEC 8859-15:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 15: Latin alphabet No. 9*

[15] ISO/IEC 8859-16:2001, *Information technology — 8-bit single-byte coded graphic character sets — Part 16: Latin alphabet No. 10*

[16] ISO/IEC 15416, *Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Linear symbols*

[17] ISO/IEC 15417, *Information technology — Automatic identification and data capture techniques — Code 128 bar code symbology specification*

[18] ISO/IEC 15424, *Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)*

[19] ISO/IEC 15434, *Information technology — Automatic identification and data capture techniques — Syntax for high-capacity ADC media*

[20] ISO/IEC/TR 29158, *Information technology — Automatic identification and data capture techniques — Direct Part Mark (DPM) Quality Guideline*

[21] *AIM ITS 97-001 International Symbology Specification - QR Code*, AIM Inc.

[22] AIM International Technical Specification, Extended Channel Interpretations:—Part 1, Identification Schemes and Protocols

[23] AIM International Technical Specification, Extended Channel Interpretations:—*Part 2, Registration Procedure for Coded Character Sets and Other Data Formats*

[24] AIM International Technical Specification, Extended Channel Interpretations:—Character Set Register

[25] *GS1 General Specifications,* GS1

[26] JIS X 0208:2012, *7-bit and 8-bit double byte coded KANJI sets for information interchange*

ICS 01.080.50; 35.040

Price based on 117 pages

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™

INTERNATIONAL STANDARD

ISO/IEC 15416

Second edition
2016-12-15

Automatic identification and data capture techniques — Bar code print quality test specification — Linear symbols

Techniques automatiques d'identification et de capture des données — Spécifications pour essai de qualité d'impression des codes à barres — Symboles linéaires

EXHIBIT
1003

Reference number
ISO/IEC 15416:2016(E)

JA2987

Contents

Page

Foreword .... **iv**
Introduction .... **v**
1 Scope **1**
2 Normative references **1**
3 Terms and definitions **1**
4 Symbols and abbreviated terms **2**
4.1 Abbreviated terms 2
4.2 Symbols 3
5 Measurement methodology **4**
5.1 General requirements 4
5.2 Reference reflectivity measurements 4
5.2.1 General 4
5.2.2 Measurement light source 4
5.2.3 Measuring aperture 4
5.2.4 Optical geometry 5
5.2.5 Inspection band 6
5.2.6 Number of scans 7
5.3 Scan reflectance profile 7
5.4 Scan reflectance profile assessment parameters 8
5.4.1 General 8
5.4.2 Element determination 9
5.4.3 Edge determination 9
5.4.4 Decode 10
5.4.5 Symbol contrast (SC) 10
5.4.6 Edge contrast (EC) 10
5.4.7 Modulation (MOD) 10
5.4.8 Defects 10
5.4.9 Decodability 12
5.4.10 Quiet zone check 13
6 Symbol grading **13**
6.1 General 13
6.2 Scan reflectance profile grading 13
6.2.1 Decode 14
6.2.2 Reflectance parameter grading 14
6.2.3 Decodability 14
6.3 Expression of symbol grade 15
7 Substrate characteristics **15**
Annex A (normative) **Decodability** **16**
Annex B (informative) **Example of symbol quality grading** **17**
Annex C (informative) **Substrate characteristics** **19**
Annex D (informative) **Interpretation of the scan reflectance profile and profile grades** **23**
Annex E (informative) **Guidance on selection of light wavelength** **26**
Annex F (informative) **Guidance on number of scans per symbol** **28**
Annex G (informative) **Example of verification report** **29**
Annex H (informative) **Comparison with traditional methodologies** **30**
Annex I (informative) **Process control requirements** **33**
Bibliography **36**

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 15416:2000), which has been technically revised with the following changes, as well as minor editorial modifications:

— the computation of "Defects" was modified in this revision of ISO/IEC 15416 (see Note 3 in 5.4.8); and

— sharp boundaries between grade levels are avoided by assigning grades within grade boundaries to the first decimal place (see the Notes in 6.2.2 and 6.2.3).

Introduction

The technology of bar coding is based on the recognition of patterns encoded in bars and spaces of defined dimensions according to rules defining the translation of characters into such patterns, known as the symbology specification.

The bar code symbol is produced in such a way as to be reliably decoded at the point of use, if it is to fulfil its basic objective as a machine readable data carrier.

Manufacturers of bar code equipment and the producers and users of bar code symbols therefore require publicly available standard test specifications for the objective assessment of the quality of bar code symbols, to which they can refer to when developing equipment and application standards or determining the quality of the symbols. Such test specifications form the basis for the development of measuring equipment for process control and quality assurance purposes during symbol production, as well as afterwards.

The performance of measuring equipment is the subject of a separate standard, ISO/IEC 15426-1.

This document is to be read in conjunction with the symbology specification applicable to the bar code symbol being tested, which provides symbology-specific detail necessary for its application.

This methodology provides symbol producers and their trading partners a universally standardized means for communicating about the quality of bar code symbols after they have been printed.

Automatic identification and data capture techniques — Bar code print quality test specification — Linear symbols

1 Scope

This document:

— specifies the methodology for the measurement of specific attributes of bar code symbols;

— defines a method for evaluating these measurements and deriving an overall assessment of symbol quality; and

— provides information on possible causes of deviation from optimum grades to assist users in taking appropriate corrective action.

This document applies to those symbologies for which a reference decode algorithm has been defined, and which are intended to be read using linear scanning methods, but its methodology can be applied partially or wholly to other symbologies.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
bar reflectance
lowest reflectance value in the scan reflectance profile of a bar element

3.2
decode
determination of the information encoded in a bar code symbol

3.3
edge contrast
difference between *bar reflectance* (3.1) and *space reflectance* (3.14) of two adjacent elements

3.4
element reflectance non-uniformity
reflectance difference between the highest *peak* (3.9) and the lowest *valley* (3.16) in the scan reflectance profile of an individual element or quiet zone

3.5
global threshold
reflectance level midway between the maximum and minimum reflectance values in a scan reflectance profile used for the initial identification of elements

3.6
inspection band
band (usually from 10 % to 90 % of the height of a bar code symbol) across which measurements are taken

Note 1 to entry: See Figure 2.

3.7
measuring aperture
opening which governs the effective *sample area* (3.10) of the symbol, and the dimensions of which at 1:1 magnification is equal to that of the sample area

3.8
modulation
ratio of minimum *edge contrast* (3.3) to *symbol contrast* (3.15)

3.9
peak
point of higher reflectance in a scan reflectance profile with points of lower reflectance on either side

3.10
sample area
effective area of the symbol within the field of view of the measurement device

3.11
scan path
line along which the centre of the *sample area* (3.10) traverses the symbol, including quiet zones

3.12
show-through
property of a substrate that allows underlying markings or materials to affect the reflectance of the substrate

3.13
space
light element corresponding to a region of a scan reflectance profile above the *global threshold* (3.5)

3.14
space reflectance
highest reflectance value in the scan reflectance profile of a space element or quiet zone

3.15
symbol contrast
difference between the maximum and minimum reflectance values in a scan reflectance profile

3.16
valley
point of lower reflectance in a scan reflectance profile with points of higher reflectance on either side

4 Symbols and abbreviated terms

4.1 Abbreviated terms

EC	edge contrast
EC_{min}	minimum value of EC
ERN	element reflectance non-uniformity
ERN_{max}	maximum value of ERN

GT global threshold

MOD modulation

PCS print contrast signal

RT reference threshold

SC symbol contrast

4.2 Symbols

A average achieved width of element or element combinations of a particular type

c defect adjustment constant

e width of widest narrow element

E width of narrowest wide element

e_i i'th edge to similar edge measurement, counting from leading edge of symbol character

F factor used to soften the effect on defect grades derived from small changes peaks and valleys within an element

K smallest absolute difference between a measurement and a reference threshold

k number of element pairs in a symbol character in a (n, k) symbology

M width of element showing greatest deviation from A

m number of modules in a symbol character

N average achieved wide to narrow ratio

n number of modules in a symbol character in a (n, k) symbology

R_b bar reflectance

R_D dark reflectance

R_L light reflectance

R_{max} maximum reflectance

R_{min} minimum reflectance

R_s space reflectance

RT_j reference threshold between measurements j and (j + 1) modules wide

S total width of a character

V decodability value

V_C decodability value for a symbol character

Z average achieved narrow element dimension or module size, as measured

5 Measurement methodology

5.1 General requirements

The measurement methodology defined in this document is designed to maximize the consistency of both reflectivity and bar and space width measurements of bar code symbols on various substrates. This methodology is also intended to correlate with conditions encountered in bar code scanning hardware.

Measurements shall be made with a defined light source (such as a single light wavelength) and a measuring aperture of dimensions defined by the application specification or determined in accordance with 5.2.1 and 5.2.2. A circular aperture is defined by its diameter in accordance with Table 1. Application specifications may define other aperture diameters or shapes.

Whenever possible, measurements shall be made on the bar code symbol in its final configuration, i.e. the configuration in which it is intended to be scanned. If this is impossible, refer to Annex C for the method to be used for measuring reflectance for non-opaque substrates.

The sampling method should be based on a statistically valid sample size within the lot or batch being tested. A minimum grade for acceptability shall be established prior to quality control inspection. In the absence of a sampling plan defined in formal quality assurance procedures or by bilateral agreement, a suitable plan may be based on the recommendations in ISO 2859-1.

5.2 Reference reflectivity measurements

5.2.1 General

Equipment for assessing the quality of bar code symbols in accordance with this document shall comprise a means of measuring and analysing the variations in the diffuse reflectivity of a bar code symbol on its substrate along a number of scan paths which shall traverse the full width of the symbol including both quiet zones. The basis of this methodology is the measurement of diffuse reflectance from the symbol.

All measurements on a bar code symbol shall be made within the inspection band defined in accordance with 5.2.4.

The measured reflectance values shall be expressed in percentage terms by means of calibration and reference to recognized national standards laboratories, where 100 % should correspond to the reflectance of a barium sulphate or magnesium oxide reference sample.

5.2.2 Measurement light source

The light source used for measurements should be specified in the application specification to suit the intended scanning environment. When the light source is not specified in the application specification, measurements should be made using the light source that approximates most closely to the light source expected to be used in the scanning process. Light sources may include narrow band or broad band illumination. Refer to Annex E for guidance on the selection of the light source.

5.2.3 Measuring aperture

The nominal diameter of the measuring aperture should be specified by the user application specification to suit the intended scanning environment. When the measuring aperture diameter is not specified in the application specification, Table 1 should be used as a guide. In an application where a range of X dimensions will be encountered, all measurements shall be made with the aperture appropriate to the smallest X dimension to be encountered.

In the absence of a defined X dimension, the Z dimension shall be substituted.

The effective measuring aperture diameter may vary slightly from its nominal dimension due to manufacturing tolerances and optical effects. Note that the measured width of some of the narrow elements may be smaller than the measuring aperture diameter.

Table 1 — Guideline for diameter of measuring aperture

X Dimension (mm)	Aperture diameter (mm)	Reference number
0,100 ≤ X < 0,180	0,075	03
0,180 ≤ X < 0,330	0,125	05
0,330 ≤ X < 0,635	0,250	10
0,635 < X	0,500	20
NOTE The aperture reference number approximates to the measuring aperture diameter in thousandths of an inch.		

NOTE The measuring aperture is not to be confused with the F-number of a lens.

5.2.4 Optical geometry

The reference optical geometry for reflectivity measurements shall consist of the following:

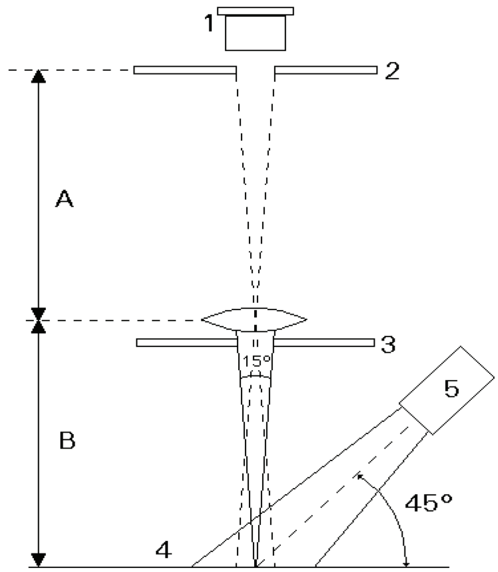
a) a source of incident illumination which is uniform across the sample area at 45° from a perpendicular to the surface, and in a plane containing the illumination source that shall be both perpendicular to the surface and parallel to the bars;

b) a light collection device, the axis of which is perpendicular to the surface.

The light reflected from a circular sample area of the surface shall be collected within a cone; the angle at the vertex of which is 15°, centred on the perpendicular to the surface, through a circular measuring aperture, the diameter of which at 1:1 magnification shall be equivalent to that of the sample area.

NOTE Figure 1 illustrates the principle of the optical arrangement, but is not intended to represent an actual device.

This reference geometry is intended to minimize the effects of specular reflection and to maximize those of diffuse reflection from the symbol. It is intended to provide a reference basis to assist the consistency of measurement. It may not correspond with the optical geometry of individual scanning systems. Alternative optical geometries and components may be used, provided that their performance can be correlated with that of the reference optical arrangement defined in this subclause.



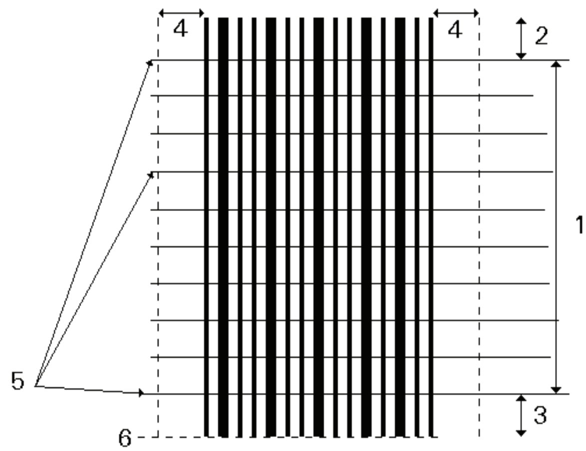
Key

1 light sensing element
2 aperture at 1:1 magnification (measurement A = measurement B)
3 baffle
4 sample
5 light source

Figure 1 — Reference optical arrangement

5.2.5 Inspection band

The area within which all measurement scan paths shall lie shall be contained between two lines perpendicular to the height of the bars of the symbol, as illustrated in Figure 2. The lower line shall be positioned at a distance above the average lower edge of the bar pattern of the symbol while the upper line shall be positioned at the same distance below the average upper edge of the bar pattern of the symbol. This distance shall be equal to 10 % of the average bar height or the measuring aperture diameter, whichever is greater. The inspection band shall extend to the full width of the symbol including quiet zones.



Key

1 inspection band (normally 80 % of average bar height)
2 10 % of average bar height, or aperture diameter if greater, above inspection band
3 10 % of average bar height, or aperture diameter if greater, above average bar bottom edge
4 quiet zones
5 scanning lines
6 average bar bottom edge

Figure 2 — Inspection band

5.2.6 Number of scans

In order to provide for the effects of variations in symbol characteristics at different positions in the height of the bars, a number of scans shall be performed across the full width of the symbol including both quiet zones with the appropriate measuring aperture and a light source of defined nominal wavelength. These scans shall be approximately equally spaced through the height of the inspection band. The minimum number of scans per symbol should normally be 10 or the height of the inspection band divided by the measuring aperture diameter, whichever is lower. Refer to Annex F for guidance on the number of scans.

The overall quality grade of the symbol is determined by averaging the quality grades of the individual scans, in accordance with Clause 6.

5.3 Scan reflectance profile

Bar code symbol quality assessment shall be based on an analysis of the scan reflectance profiles. The scan reflectance profile is a plot of reflectance against linear distance across the symbol. If scanning speed is not constant, measuring devices plotting reflectance against time should make provision to compensate for the effects of acceleration or deceleration. If the plot is not a continuous analogue profile, the measurement intervals should be sufficiently small to ensure that no significant detail is lost and that dimensional accuracy is adequate.

Figure 3 is a graphical representation of a scan reflectance profile. The vertical axis represents reflectance and the horizontal axis linear position. The high-reflectance areas are spaces and the low-reflectance areas are bars. The high-reflectance areas on the extreme left and right are the quiet zones. The important features of the scan reflectance profile can be determined by manual graphical

analysis or automatically by numerical analysis. For example, the highest reflectance point on the scan reflectance profile in Figure 3 is approximately 82 % and the lowest is approximately 10 %.

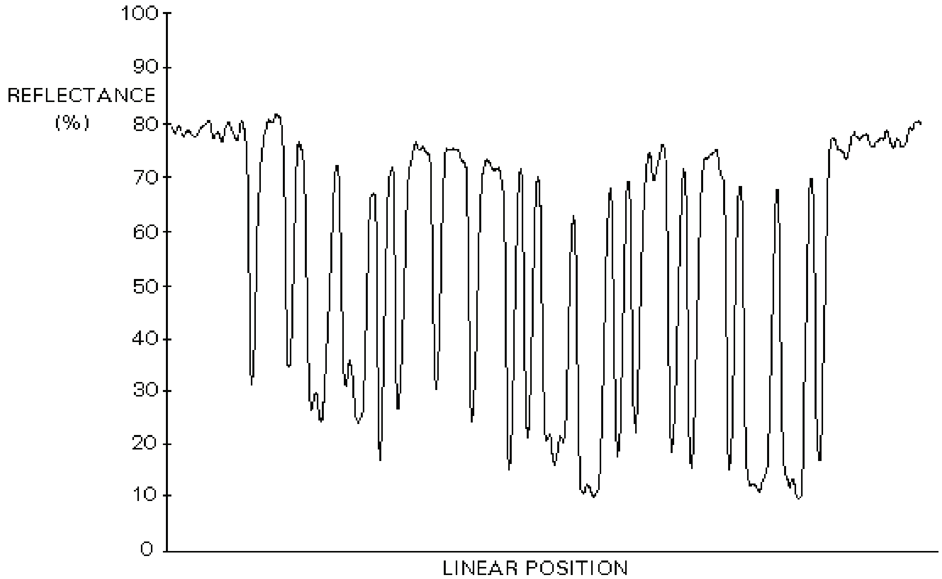


Figure 3 — Scan reflectance profile

5.4 Scan reflectance profile assessment parameters

5.4.1 General

The scan reflectance profile parameters described in 5.4.2 to 5.4.9 shall be assessed for compliance with this document. Grading of the scan reflectance profile parameters is described in 6.2. Figure 4 is the same scan reflectance profile as Figure 3 with certain features indicated.

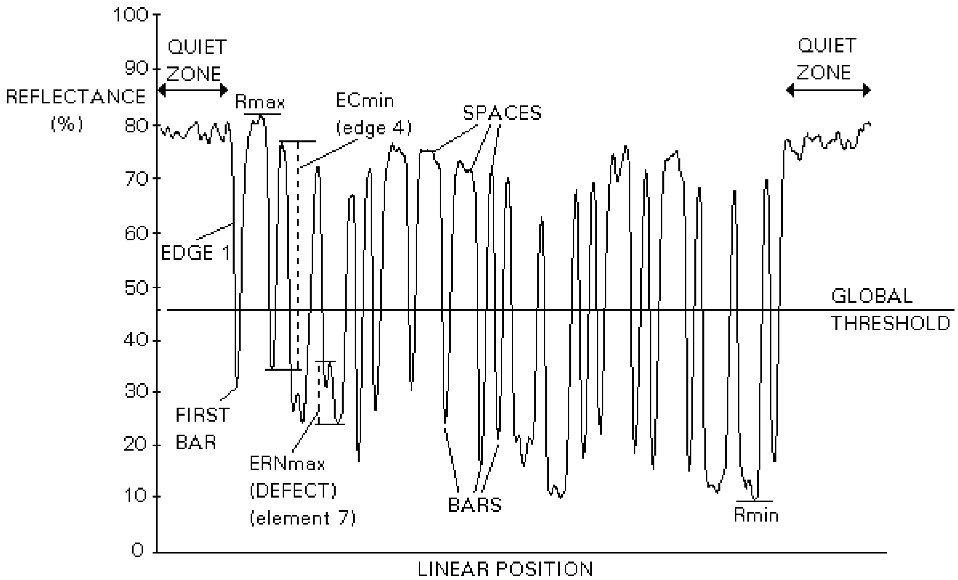


Figure 4 — Features of scan reflectance profile

5.4.2 Element determination

To locate the bars and spaces, a global threshold shall be established. The global threshold shall be the reflectance value midway between the highest and lowest reflectance values measured in the scan reflectance profile, or:

$$GT = (R_{max} + R_{min})/2$$

where

R_{max} is the highest reflectance value;

R_{min} is the lowest reflectance value.

Each region above the global threshold shall be regarded as a space and the highest reflectance value in the region shall be designated the space reflectance, R_s. Similarly, the region below the global threshold shall be regarded as a bar and the lowest reflectance in the region shall be designated the bar reflectance, R_b.

For each space, R_s – GT represents its reflectance margin above the global threshold. For each bar, GT – R_b represents its reflectance margin below the global threshold. A warning should be issued when the minimum reflectance margin for any element is less than 5 % of the SC of a symbol. This warning should caution users to consider the possibility that this symbol is close to an F grade for edge determination.

NOTE This warning is not required and this recommendation is newly introduced in this revision of this document.

5.4.3 Edge determination

An element edge shall be defined as being located at the point where the scan reflectance profile intersects the mid-point between R_s and R_b of two adjacent regions, i.e. where the reflectance value is $(R_s + R_b)/2$. If more than one point satisfying this definition exists between adjoining elements, then

the edge position and the element widths will be ambiguous and the scan reflectance profile shall fail the decode parameter. The quiet zones and intercharacter gaps, if any, are considered to be spaces.

5.4.4 Decode

The symbology reference decode algorithm shall be used to decode the symbol using the element edges determined in 5.4.3. This algorithm may be found in the symbology specification.

5.4.5 Symbol contrast (SC)

Symbol contrast is the difference between the highest and lowest reflectance values in a scan reflectance profile.

$$SC = R_{max} - R_{min}$$

5.4.6 Edge contrast (EC)

Edge contrast is the difference between the R_s and R_b of adjoining elements including quiet zones. The lowest value of edge contrast found in the scan reflectance profile is the minimum edge contrast, EC_{min}.

$$EC = R_s - R_b$$

5.4.7 Modulation (MOD)

Modulation is the ratio of the minimum edge contrast to symbol contrast.

$$MOD = EC_{min}/SC$$

5.4.8 Defects

Defects are irregularities found within elements and quiet zones, and are measured in terms of element reflectance non-uniformity.

Element reflectance non-uniformity within an individual element or quiet zone is the difference between the reflectance of the highest peak and the reflectance of the lowest valley. When an element consists of a single peak or valley, its reflectance non-uniformity is zero. The highest value of element reflectance non-uniformity found in the scan reflectance profile is the maximum element reflectance non-uniformity. Defect measurement is expressed as the ratio of the maximum element reflectance non-uniformity (ERN_{max}) to symbol contrast.

a) Define the defect adjustment constant “c” equal to 0,075.

   NOTE 1 “c” corresponds to the following:

   — a small amount of “noise” to be reduced to eliminate instability in measurement;

   — an amount of contrast difference that is small enough for scanners to ignore.

   NOTE 2 If “c” would be defined as 0, the method described here is equivalent to the defect grade specified in the previous edition of this document in all cases (because the factor, F, defined below, would always be equal to 1).

b) For each bar element.

   1) For each positive Peak Maxima in the element:

      i) find the lowest valley to the left of it within the element, called $R_{minLeft}$;

      ii) find the lowest valley to the right of it within the element, called $R_{minRight}$;

      iii) calculate ERN_{left} as the Peak Maximum − $R_{minLeft}$;

iv) calculate ERN_{right} as the Peak Maximum – $R_{minRight}$;

v) take the lesser of ERN_{left} and ERN_{right} as ERN′ (ERN prime);

vi) set F to the value 1 if ERN′ ≥ c. If ERN′ < c, then calculate F = ERN′/c;

vii) calculate the provisional ERN for this peak (and only this peak) as F × MAX (ERN_{left}, ERN_{right}).

2) Take the maximum of the provisional ERN values from all iterations of the previous step as the ERN of this element.

c) Same as b) for each space element, and as follows.

1) For each negative Valley Minima (a local minima):

i) find the highest peak to the left of it within the element, called $R_{maxLeft}$;

ii) find the highest peak to the right of it within the element, called $R_{maxRight}$;

iii) calculate ERN_{left} as $R_{maxLeft}$ – the Valley minimum;

iv) calculate ERN_{right} as $R_{maxRight}$ – the Valley minimum;

v) take the lesser of ERN_{left} and ERN_{right} as ERN′ (ERN prime);

vi) set F to the value 1 if ERN′ ≥ c. If ERN′ < c, then calculate F = ERN′/c;

vii) calculate the provisional for this valley (and only this valley) as F × MAX (ERN_{left}, ERN_{right}).

2) Take the maximum of all the provisional ERN values from all iterations of the previous step as the ERN of this element.

d) Take the maximum of all ERN values from b) 2) and c) 2) as ERN_{max} for the overall scan.

Defects = ERN_{max}/SC

NOTE 3 The calculation of ERN_{max} described above is modified in this revision of this document.

Three cases are especially useful to illustrate the functioning of this algorithm. The leftmost example shown in Figure 5 is an example of a case that will be affected by this change. The defect will be reduced because ERN_{left} is very small (in particular, it is much less than “c”). The middle example shows a case where many peaks and valleys exist within an element, but ERN_{left} and ERN_{right} are much larger than “c”. The defect measurement will not be affected by this change. The rightmost example is actually equivalent to the middle example in as much as this algorithm is concerned, even though ERN_{left} and ERN_{right} are different for each local maxima.

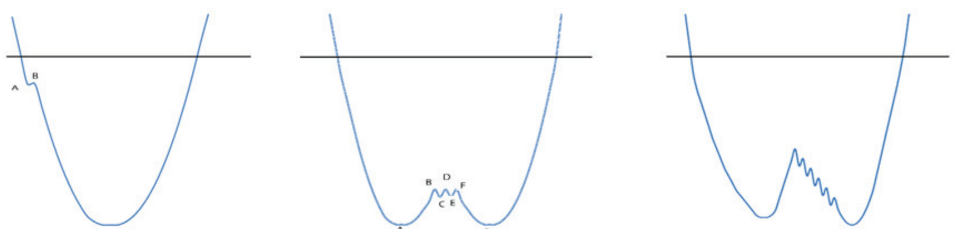


Figure 5 — Examples to illustrate ERN calculation

5.4.9 Decodability

The decodability of a bar code symbol is a measure of the accuracy of its production in relation to the appropriate reference decode algorithm. Bar code scanning equipment can generally be expected to perform better on symbols with higher levels of decodability than on those with lower decodability.

Rules governing the nominal dimensions for each bar code symbology are given in particular symbology specifications. The reference decode algorithm allows reasonable margin for errors in the printing and reading processes by defining one or more reference thresholds at which a decision is made as to the widths of elements or other measurements.

The decodability of a scan reflectance profile is the fraction of available margin which has not been consumed by the printing process and is thus available for the scanning process. When calculating the decodability value, V, for a scan reflectance profile, regard shall be to the measurements required by the reference decode algorithm in the relevant symbology specification. In the following paragraph, the term “measurement” shall be taken to refer to either to a single element width, in symbologies which use these directly in the reference decode algorithm (e.g. “Code 39”), or to the combined width of two or more adjacent elements, in symbologies using edge to similar edge measurements for decoding (e.g. “Code 128”).

The decodability value is calculated with reference to the following:

a) the average achieved width (referred to in the formula below as A) for measurements of a particular type [e.g. narrow elements, or bar + space combinations nominally totalling 2 (or 3, or 4 ...) modules] in the scan reflectance profile;

b) the reference threshold applicable to measurements of the same type as A (referred to in the formula below as RT);

c) the actual measurement showing the greatest deviation from A in the direction of the reference threshold, (referred to in the formula below as M).

The general form of the formula for calculating V is as follows:

$$V = \text{absolute value of } [(RT - M)/(RT - A)]$$

where

(RT – M) is the remaining margin not used by printing variation;

(RT – A) is the total theoretical margin based on the ideal measurement of the element(s).

Figure 6 illustrates this principle. The shaded area represents the range of measurements of the same type as A (e.g. narrow elements). All measurements are taken from 0.

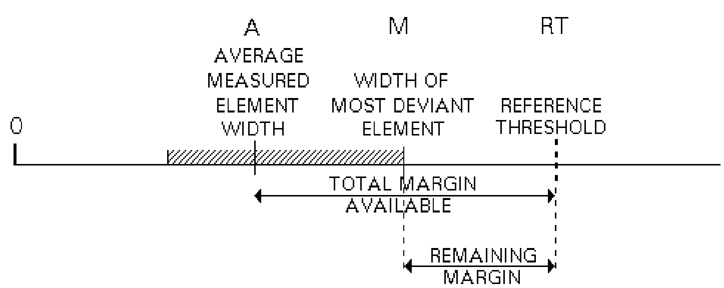


Figure 6 — Principle of decodability measurement

JA3004

More specific formulae applicable to either two-width symbologies or (n, k) symbologies are defined in Annex A. Reference should also be made to the symbology specification for the particular computation of decodability unique to each symbology.

5.4.10 Quiet zone check

The average narrow element width, Z, shall be calculated and revised quiet zones determined based on this dimension. R_{max}, ERN of the quiet zones and R_s of the quiet zones, as used in the initial scan reflectance profile analysis, shall be compared with new values obtained for the revised quiet zones. If the value(s) differ, then affected portions of the scan reflectance profile analysis shall be repeated.

6 Symbol grading

6.1 General

As a consequence of the use of different types of bar code reading equipment under differing conditions in actual applications, the level of quality required of a bar code symbol to ensure an acceptable level of performance will differ. Application specifications should therefore define the required performance in terms of symbol grade in accordance with this document, following the guidelines in D.3.

Symbol grading shall be used to derive a relative measure of symbol quality under the measurement conditions used. Each scan reflectance profile shall be analysed and assigned a grade on a descending scale from 4 to 0 in steps of 0,1. The grade 4 represents the highest quality, while the grade 0 represents failure. The scan reflectance profile grade for each scan reflectance profile shall be the lowest grade of any parameter in that scan reflectance profile. The overall symbol grade shall be the arithmetic mean of the scan reflectance profile grades. If any two scans of the same symbol yield different decoded data, then the overall symbol grade, irrespective of individual scan reflectance profile grades, shall be 0. An example of a symbol quality grading is given in Annex B. For the interpretation of the scan reflectance profile and profile grades, see Annex D.

In order to determine the causes of poor quality grades, it is necessary to examine the grades for each parameter in the scan reflectance profile in question as described in D.2. For process control purposes, the averages of the grade for each parameter obtained from all the scan reflectance profiles may provide meaningful guidance (see I.4). If the grades alone do not provide sufficient explanation, it may also be necessary to examine the plot(s) of the scan reflectance profile(s).

6.2 Scan reflectance profile grading

The scan reflectance profile grade shall be the lowest grade of the following:

a) decode;

b) symbol contrast (SC);

c) minimum reflectance (R_{min});

d) minimum edge contrast (EC_{min});

e) modulation (MOD);

f) defects;

g) decodability (V);

h) any additional requirements imposed by the application or symbology specification.

It is appropriate to measure these parameters in the sequence given above.

6.2.1 Decode

Decodable symbols shall comply with the symbology specification, notably in respect of character encodation, start and stop patterns, symbol check character(s), quiet zones and intercharacter gaps (where applicable). If the scan reflectance profile cannot be decoded using the symbology reference decode algorithm, then it shall receive the failing grade 0. Otherwise, it shall receive the grade 4.

6.2.2 Reflectance parameter grading

Depending on their values, symbol contrast, modulation and defects may be graded from 4,0 to 0 in steps of 0,1; minimum reflectance and minimum edge contrast grades may be graded either 4 or 0. These parameters are interdependent and need to be considered together.

Table 2 defines the parameter values corresponding to the various grades.

Table 2 — Reflectance parameter grading

Grade	R_{min}	SC	EC_{min}	MOD	Defects
4,0	≤0,5 R_{max}	≥70 %	≥15 %	≥0,70	≤0,15
3,0		≥55 %		≥0,60	≤0,20
2,0		≥40 %		≥0,50	≤0,25
1,0		≥20 %		≥0,40	≤0,30
0	>0,5 R_{max}	<20 %	<15 %	<0,40	>0,30

For SC, MOD and defects, the grade shall be computed as an interpolated value, rounded to the nearest 0,1 in between grade levels. For example, SC of 52 % shall result in a grade of 2,8 and MOD of 0,69 shall results in a grade of 3,9. In the lowest range, the grade shall be interpolated from 1 down to 0, except defect which shall be 0 for all values greater than 0,30. The decimal part of the SC grade is computed as the fraction of the range for the grade of 2 (15 %) that the measured value (52 %) exceeds the minimum value for a grade of 2 (40 %), computed as 2 + [(52 % − 40 %)/15 %].

NOTE The interpolation described above is a new feature in this revision of this document and is introduced as a way of reducing meaningless grade level fluctuations when small changes in measurements cause a grade to transition between grade levels.

6.2.3 Decodability

The decodability value, V, for each scan reflectance profile shall be calculated according to the formula for the type of symbology in question set out in Annex A, supplemented where necessary by formulae specific to the symbology in question, contained in the symbology specification. Decodability is graded from 4 to 0, rounded to the nearest 0,1, in between grade levels according to Table 3. For example, for V of 0,56 shall result in a grade of 3,5 and V of 0,20 shall result in a grade of 0,8.

Table 3 — Decodability grades

V	Grade
≥0,62	4
≥0,50	3
≥0,37	2
≥0,25	1
<0,25[a]	0

[a] For values less than 0,25, interpolate from 1 down to 0.

NOTE The interpolation described above is a new feature in this revision of this document and is introduced as a way of reducing meaningless grade level fluctuations when small changes in measurements cause a grade to transition between grade levels.

6.3 Expression of symbol grade

A symbol grade is only meaningful if it is expressed in conjunction with the measurement light source and aperture used. It should be shown in the format G/A/L, where G is the overall symbol grade, i.e. the arithmetic mean of the scan reflectance profile grades to one decimal place, A is the aperture reference number, from Table 1, and L indicates the light source, by the light peak wavelength in nanometres for narrow band illumination, the letter "W" for white (broad band) illumination, or other designator defined by an application specification.

For example, 2,7/05/660 would indicate that the average of the grades of the scan reflectance profiles was 2,7 when these scan reflectance profiles were obtained with the use of a 0,125 mm aperture (ref. no. 05) and a 660 nm light source.

7 Substrate characteristics

Certain characteristics of the substrate, notably gloss, low opacity and the presence of an over-laminate may affect reflectance measurements, and the recommendations in Annex C should be taken into account if any of these factors is present.

Annex A
(normative)

Decodability

A.1 General

This annex defines general formulae for the calculation of the decodability value, V, for symbologies for which the reference decode algorithm defines reference thresholds. These formulae may be supplemented by additional formulae specific to an individual symbology and defined in the relevant symbology specification.

A.2 Two-width symbologies

In each scan reflectance profile, calculate Z and N for the whole symbol.

For each symbol character or auxiliary pattern, calculate RT in accordance with the reference decode algorithm.

Then,

$V1 = (RT - e)/(RT - Z)$

$V2 = (E - RT)/[(N \times Z) - RT]$

V_C = the lesser of V1 or V2

The decodability value, V, for the scan reflectance profile shall be the lowest value of V_C for any symbol character or auxiliary pattern.

A.3 Edge to similar edge decodable symbologies [(n, k) symbologies]

If necessary in each scan reflectance profile, calculate Z for the whole symbol:

$Z = (\text{average } S)/n$

where S and n are as defined in 4.2.

For each symbol character, determine a set of reference thresholds RT_j:

for all $j = 1$ to $n - 2(k - 1)$

$RT_j = [(j + 0,5) \times S]/n$

where S, n and k are as defined in 4.2.

for all $i = 1$ to $2(k - 1)$ and all $j = 1$ to $n - 2(k - 1)$

let K = smaller of absolute value of $(e_i - RT_j)$ or previous K

where e_i is the measurement from leading edge of element i to leading edge of element (i + 2).

Then, $V_C = K/(S/2n)$.

The decodability value, V, for the scan reflectance profile shall be the lowest value of V_C for any symbol character or auxiliary pattern.

Annex B
(informative)

Example of symbol quality grading

B.1 Individual scan reflectance profile grading

This annex illustrates the determination of the grades for the scan reflectance profile shown in Figure 3, assuming measurement using a 900 nm (infrared) light source and a 0,125 mm aperture.

Referring to Figure 3, the actual reflectance values may be determined graphically in order to grade the scan reflectance profile.

Minimum reflectance (R_{min}) is 10 % while the maximum reflectance (R_{max}) is 82 %. The global threshold is therefore 46 %. R_{min} satisfies the (0,5 × R_{max}) test by being less than (0,5 × 82 %) = 41 %.

Symbol contrast (SC) is 82 − 10 = 72.

Minimum edge contrast (EC_{min}) occurs on edge 4, where R_s and R_b are 76 % and 34 %, respectively. EC_{min} is 76 − 34 = 42.

Modulation (MOD) is therefore 42/72 = 0,58

Maximum element reflectance non-uniformity (ERN_{max}), the largest non-uniformity or defect in a scan reflectance profile, can be found as the result of a void in element 7, a bar. ERN_{max} is equal to 36 − 24 = 12. Note that the ERN_{max} can be in any bar, space or quiet zone. The defects value is therefore

12/72 = 0,17.

Assuming that the symbol has decoded correctly (as characters "Start $ M Stop" in "Code 39") and that the decodability value, V, has been calculated as 0,58, the following individual parameter grades and the scan reflectance profile grade can be determined for the scan reflectance profile in Figure 3 (see Table B.1).

Table B.1 — Grades for the scan reflectance profile as shown in Figure 3

Parameter	Value	Grade
Decode		4
R_{max}	82 %	
R_{min}	10 %	4,0
SC	82 − 10 = 72 %	4,0
EC_{min}	76 − 34 = 42 %	4,0
MOD	42/72 = 0,58	2,8
Defects	12/72 = 0,17	3,6
Decodability	0,58	3,7

Since the lowest individual grade, in this instance the grade for MOD, is 0,28, the scan reflectance profile grade is also 2,8.

See also Annex G.

B.2 Overall symbol grade

Assuming that a series of 10 scans of the symbol used in Figure 3 gave the following scan reflectance profile grades:

2, 2, 3, 3, 4, 2, 2, 2, 3, 3,

the arithmetic mean of these grades, and hence the overall symbol grade, is 2,6. The result should be reported in the form

2,6/05/900.

For information, this result would be shown, using alphabetic grading as

B/05/900.

Annex C
(informative)

Substrate characteristics

C.1 General

In certain circumstances (for example, the design and production of printed packaging materials incorporating bar code symbols), it is necessary or desirable to assess the acceptability of substrates and/or ink colours for a given bar code application, before a bar code symbol is available, which could be tested in accordance with this document.

C.2 Substrate opacity

The symbol shall be graded according to the reflectance parameters in 6.1.2 when measured in its final configuration, e.g. final filled package.

If it is not possible to measure the symbol in this configuration, then the effects of show-through of high-contrast interfering patterns may be ignored if when measured as follows, the substrate opacity is 0,85 or greater. If the opacity is less than 0,85, the symbol should be measured when backed by a uniform dark surface the reflectance of which is not more than 5 %.

The opacity of the substrate shall be calculated as follows:

$$\text{Opacity} = R2/R1$$

where

R1 is the reflectance of a sample sheet of the substrate backed with a white surface the reflectance of which is 89 % or greater;

R2 is the reflectance of the same sample sheet backed with a black surface of not more than 5 % reflectance.

C.3 Gloss

The reference illumination conditions specified for the measurement of reflectance should enable the maximum rejection of specular reflection while giving a representative assessment of the diffuse reflectance of the symbol and substrate. Highly glossy materials and those whose diffuse reflectance characteristics vary with the angle of incident and/or collected light may yield grades differing from those obtained by the use of the reference optical arrangement.

C.4 Over-laminate

A symbol intended to be covered with a protective lamination should be graded according to the reflectance parameters in 6.2.2 when measured with the laminate in place. The thickness of the laminate including its adhesive should be as small as possible in order to minimize its effects on the reading performance of the symbol.

C.5 Static reflectance measurements

In some cases, it may be desirable to carry out static reflectance measurements of samples of the substrate on which a bar code is to be printed and on colour patches or ink samples which replicate the colour in which the bar code will be printed. The following guidelines provide a means which, if it is followed, will predict as closely as is generally possible, the results which will be obtained when the symbol is scanned dynamically.

Static reflectance measurements should be made with the wavelength of light, aperture size and optical arrangement which relate to the application and which are specified in accordance with 5.2.1 to 5.2.3.

Where reflectance measurement equipment meeting the requirements of this annex is not available, optical density measurements may be made using a standard densitometer with an appropriate light source and converted to reflectance values; density (D) and reflectance (R) are related as follows:

$$R = 100/10^{D}.$$

NOTE It is impossible to predict to a high degree of accuracy the symbol contrast and, in particular, the edge contrast which will be achieved in the printed symbol. It is therefore appropriate to allow some safety margin above the minimum values for specified grades.

C.5.1 Prediction of symbol contrast (SC)

The prediction of SC requires that measurements of reflectance be made on samples which simulate the highest (R_{max}) and lowest reflectance (R_{min}) areas which will be present in the finished symbol.

It is probable that in most bar code symbols, R_{max} will be found in the quiet zone of the symbol; therefore to simulate the conditions found in the quiet zone, R_{max} should be measured in the centre of a sample area, at least 10× in diameter, of the material on which the symbol is to be printed.

It is probable that in most bar code symbols, R_{min} will be found in the widest bars of the symbol; therefore to simulate the conditions most likely to yield a value of R_{min} consistent with that which would be found in practice, reflectance should be measured in the centre of a strip of material 2× to 3× wide and which matches the colour in which the bars are to be printed.

A predicted value of SC can then be calculated as follows:

$$SC' = R_{max} - R_{min}$$

C.5.2 Prediction of minimum edge contrast (EC_{min}) and modulation (MOD)

In order to assess the grade for modulation (MOD), it is necessary to predict the minimum value of edge contrast likely to be found in practice. It is best to make measurements of edge contrast on the printed symbol. If that is not possible, the prediction of EC_{min} requires that measurements of reflectance be made on samples which simulate the smallest reflectance difference which will be found between adjacent elements. It is probable that in most bar code symbols, this condition will be found where a light and a dark element which are each 1× in width are adjacent to each other and where the element on the other side of the light element is a wide dark element.

To simulate this condition, a sample of material, which is of the colour in which the bar code symbol will be printed, should be cut to form a mask of the type shown in Figure C.1.

Figure C.1 — Mask for static reflectance measurements

The mask shown in Figure C.1 should be made of a material that is as thin as is practical. It will however have some thickness and would therefore be capable of casting a shadow. To ensure that the effects of this are minimized, it is essential that the light source(s) of the instrument used to make the measurements are oriented to be in line with the long axis of the elements in which the measurements are being made. The narrow dark element AA and the narrow light element BB should each be equal in width to the X dimension of the symbol to be printed and the height of BB should be at least 20× or 10 mm, whichever is greater.

The measurement of the reflectance value R_s should be made in the narrow light element which is formed when the mask in Figure C.1 is placed over a background of the material and colour on which the bar code is to be printed.

The measurement of the reflectance value R_b should be made in the narrow dark element which is formed when the mask in Figure C.1 is placed over a background of the material and colour on which the bar code is to be printed.

A predicted value of Ec_{min} can then be calculated as follows:

$$Ec_{min}' = R_s - R_b$$

For materials which do not satisfy the tests for opacity, which are detailed in C.1, the measurements which are made for the purpose of predicting SC and ECmin should be made with the test samples backed by a uniform dark surface, the reflectance of which is not more than 5 %. The same measurements should then be made with the test samples backed by a uniform surface the reflectance of which is not less than 89 %. The calculated values of static SC and ECmin shall be equal to or greater than the minimum values for the grade selected for the application, for tests on both the dark and light backgrounds.

A predicted value of MOD can be calculated as follows:

$$MOD' = Ec_{min}'/SC'$$

C.5.3 Acceptability of measured and derived values

The grades corresponding to the static values of SC and EC_{min} and to the derived value for modulation (MOD) shall all be equal to or higher than the minimum overall symbol grade specified for the application.

For applications where print contrast signal (PCS) is the preferred method of determining the reflectance characteristics of a bar code symbol, an approximation of the value of PCS can be determined from the values measured for the purpose of predicting SC. Refer to Annex H.

Annex D
(informative)

Interpretation of the scan reflectance profile and profile grades

D.1 Significance of scan reflectance profiles

The scan reflectance profile represents the signal from a typical bar code scanning device. In a bar code reader, this signal is processed by an edge finding circuit prior to arriving at the decoder.

In order to allow a variety of edge finding circuits to find the intended elements, the following reflectance parameters should be considered:

— the global threshold should be traversed by every edge in the symbol;

— symbol contrast, modulation and minimum edge contrast should not be too low;

— defects and minimum reflectance should not be too high.

In addition, to allow a decoder to function, the following parameters should be considered:

— decode;

— decodability.

D.2 Interpretation of results

When examining a symbol with a view to drawing conclusions about the possible causes of low grades, individual parameter grades should be examined, as well as the overall grade. There is a degree of interdependence between the parameters, but typical causes and effects are listed below. For process control purposes, significant additional information may be derived by averaging the grades obtained for each parameter for all scan reflectance profiles. In particular, the measurement of the average bar width gain or loss may be used for monitoring the performance of a printer or printing press during an extended print run.

Bar width gain:

— may be reported directly (as average);

— reduces EC;

— reduces MOD;

— reduces decodability:

  — if not systematic, decodability will suffer though average bar width gain does not appear excessive;

  — if systematic, decodability will appear low and average bar width gain will be higher;

— may cause decode failure if excessive.

Bar width loss:

— may be reported directly (as average);

— increases EC initially; when excessive, reduces EC;

— increases MOD initially; when excessive, reduces MOD;

— may increase R_{min};

— reduces decodability:

  — if not systematic, decodability will suffer though average bar width loss does not appear excessive;

  — if systematic, decodability will appear low and average bar width loss will be higher;

— may cause decode failure if excessive.

Irregular element edges:

— cause variations in decodability between scan reflectance profiles;

— may cause decode failure if excessive.

Uneven inking:

— decreases EC;

— decreases MOD;

— may increase ERN_{max};

— may cause spurious elements to be detected (decode failure).

Voids and/or specks:

— increase ERN;

— if excessive in size may cause spurious elements to be detected (decode failure);

— may cause edge determination failure.

D.3 Matching grades to applications

Because of the varying features of bar code systems, notably:

— vertical redundancy;

— tolerances in decoding algorithms;

— ability of operators to rescan in the event of failure to read;

— availability of scanning equipment with multiple scan paths.

Symbols with differing grades may give good performance in practice. Application specifications should specify the minimum acceptable grade (together with aperture size and shape and light wavelength or light source) to suit the characteristics of the scanning environment.

Symbols with an overall grade of 3,5 or better are the best quality and will in principle perform most reliably. This grade should be specified as a minimum where the reader crosses the symbol once only (with little possibility of rescanning in the event of failure to read) or is limited to a fixed single scan path.

A symbol graded between 2,5 and 3,5 if scanned in a single path may require rescanning to decode. A minimum grade of 2,5 is appropriate for systems where the symbol will be read on most occasions in a single scan pass, but which allow for rescanning.

Symbols graded between 1,5 and 2,5 are more likely to require rescans than those with higher grades. For best read performance, devices which provide for multiple scan paths across the symbol should be used or the system should be prepared to allow frequent rescan attempts.

Symbols with grades between 0,5 and 1,5 should be read by equipment providing for multiple, unique scan paths across the symbol. Some readers may fail to scan some such symbols successfully. System designers may therefore wish to provide for alternative means of data entry in such an event. Prior to the acceptance of symbols of this grade for a particular application, it is recommended that the symbols should be tested with the type of bar code reader to be used to determine that the results are within acceptable limits.

Symbols graded below 0,5 will have had a high proportion of “failed” scan reflectance profiles and are unlikely to perform reliably with any reading equipment.

D.4 Alphabetic grading

In certain application specifications, grades are identified using the letters A, B, C, D and F to correspond to the numeric grades 4, 3, 2, 1 and 0 respectively used in this document.

Overall symbol grading using this scheme is in accordance with Table D.1.

Table D.1 — Overall symbol grading — Numeric and alphabetical grading equivalence

Numeric range	Alphabetic grade
3,5 to 4,0	A
2,5 to <3,5	B
1,5 to <2,5	C
0,5 to <1,5	D
below 0,5	F

Annex E
(informative)

Guidance on selection of light wavelength

E.1 General

5.1 and 5.2.2 require measurements to be made using the wavelength of light which the intended scanning environment will use. If, as may happen, an application specification does not specify the light source, a judgment needs to be made in order to determine the most likely wavelength, to enable valid measurements to be made and to be sure that the results will be properly indicative of likely scanning performance in the application.

E.2 Light sources

Light sources for bar code scanning applications normally fall into two broad areas, namely visible light and infrared light, although a very few highly specialized applications may call for light sources of unusual characteristics such as ultra-violet for fluorescent bar code symbols.

Visible light scanning normally uses light sources with a peak wavelength in the red part of the spectrum, between 620 nm and 700 nm. Infrared scanning uses sources with peak wavelengths between 720 nm and 940 nm.

The most common light sources used for bar code scanning are as follows:

a) helium-neon laser (633 nm);

b) light-emitting diode (numerous wavelengths, both visible and infrared);

c) solid-state laser diode (numerous wavelengths, both visible and infrared);

d) incandescent lamp (nominally white light);

e) white LED.

The key characteristics of these are as follows.

A **helium-neon laser** is a gas-filled laser tube which emits highly monochromatic coherent light at a peak wavelength of 632,8 nm (most usually rounded to 633 nm), in the visible red area of the spectrum.

A **light-emitting diode** is a low-power solid-state component most frequently found as the light source in a light pen (wand) or CCD scanner. Operating wavelengths in the visible spectrum may be from 620 nm to 680 nm; most commonly either 633/640 or about 660 nm. In the infrared spectrum, 880 nm to 940 nm is the most common range of wavelengths.

Typical wavelengths used by **solid-state laser diodes** at the date of publication of this document are 780 nm (infrared) and, in the visible spectrum, 660 nm and 680 nm. They are frequently found in hand-held (laser) scanning equipment and a number of fixed scanners.

In bar code scanning applications, **incandescent lamps** are mainly found in systems using CCD array camera and image processing technology rather than scanning techniques. The light source has a power distribution covering much of the visible spectrum and well into the infrared spectrum; optical characteristics are defined in colour temperature terms rather than in those of peak wavelength, because of the wide bandwidth and relative absence of peaks in the power distribution. When used in conjunction with a Wratten 26 filter, the light characteristics of a 2856°K lamp approximate to those of a 620 nm to 633 nm source.

White LEDs emit a combination of wavelengths that are prominent in the blue and yellow regions. The colour spectrum of white LEDs should be defined within an application.

NOTE Wavelengths stated above can change as the technology evolves.

E.3 Effect of variations in wavelength

The reflectance of a substrate or bar code symbol element varies with the wavelength of the incident light. A black, blue or green printed area will tend to absorb visible red light strongly (and appear therefore of low-reflectance), whereas a white, red or orange area will reflect most of the incident light. In the infrared spectrum, the apparent colour of the element does not correlate at all with reflectance; it is the nature of the pigmentation used (for example, the proportion of carbon content) which governs reflectance. Taking reflectance measured at 633 nm as a reference, when measured at 660 nm or 680 nm, the results may differ significantly and sufficiently to cause the symbol grade to change by one or two units, or even more in the case of bars printed on some thermal papers.

Annex F
(informative)

Guidance on number of scans per symbol

Bar code symbols are designed to provide a considerable degree of vertical redundancy of the information contained in them. Localized defects and variations in symbol characteristics may occur in the height of the symbol, resulting in the likelihood of scan reflectance profiles from different scan paths across the symbol differing significantly. It is therefore necessary to assess the overall symbol quality by averaging scan reflectance profile grades from multiple scan paths.

The minimum number of scans per symbol as defined in 5.2.5 should normally be 10 or the height of the inspection band divided by the measuring aperture diameter, whichever is lower.

Where the production process (in particular, in the circumstances defined in I.1) has been shown to be subject to a relatively low incidence of the defects and variations referred to above through documented formal quality assurance procedures in accordance with ISO 9000 and related standards, the number of scans per symbol may be reduced in order to simplify the process of assessment of large numbers of symbols. Refer to I.2 for details of this reduction.

Annex G
(informative)

Example of verification report

There exists a wide range of verification equipment designed to measure the quality of bar code symbols. Table G.1 illustrates an example report produced by one of these devices (assuming that the report below was obtained with the use of a measuring aperture of 0,250 mm diameter (ref. no. 10) and with a 660 nm light source. The grade should therefore be reported as 3,3/10/660.

Table G.1 — Example of verification report

VERIFICATION REPORT			
Date	23.12.14	Time	16:12:36
Aperture:	0,25 mm	Wavelength:	660 nm
Symbology:	Code 39	Decoded data:	$M
Overall Symbol Grade:	3,3 (B)	Averaged over (no. of scans):	1
Scan reflectance profile analysis			
Parameter	**Value**	**Grade**	
Decode	Pass	4	
R_{max}	79 %	N/A	
R_{min}	2 %	4	
Global threshold	41 %	N/A	
Symbol contrast	77 %	4,0	
Min. edge contrast	48 %	4	
Modulation	63 %	3,3[a]	
Defects	16 %	3,8	
Decodability	75 %	4,0	
PCS	97 %	N/A	
Average bar gain	+3,0 %	N/A	
[a] Parameter grade(s) determining scan reflectance profile grade.			

Annex H
(informative)

Comparison with traditional methodologies

H.1 Traditional methodologies

Traditionally, two methodologies have been used to assess print quality in certain application standards. Advice is given in this document to assist users, particularly producers of symbols, to compare the results obtained with these traditional parameters, which are the following:

a) the measurement of bar element widths and particularly the amount of gain or loss from the nominal element dimensions;

b) the calculation of a print contrast signal (PCS) value from the reflectance values R_L and R_D.

Where the symbols are used in an application which does not specify print quality in terms conforming with this document, these two parameters may be measured as part of the procedure to assess symbol quality and should be measured especially for the purposes of process control in symbol production (see Annex I). However, they are excluded from the grading scheme of this document because the criteria for acceptance or failure which they use do not reflect the behaviour of scanning systems. Their optional inclusion as measured, but ungraded, parameters is to enable historical quality information to be correlated with the methodology specified herein.

H.2 Correlation of print contrast signal with symbol contrast measurements

The specifications of a number of bar code applications provide for the contrast between bars and spaces or background to be assessed in terms of print contrast signal (PCS); these specifications define a minimum value of PCS for acceptability. In some cases, this is a fixed value (e.g. PCS_{min} = 0,75 is a commonly specified value); in others, PCS_{min} is itself a function of the background reflectance.

Print contrast signal is calculated according to the following formula:

$$PCS = (R_L - R_D)/R_L$$

where

R_L is the background (space) reflectance;

R_D is the bar reflectance.

Many of the specifications referred to above do not define the points at which R_L and R_D are measured. There is therefore a risk of inconsistency in the value determined for PCS. Furthermore, the profile evaluation techniques defined in this document more closely represent the nature of bar code scanning than do methods based on PCS. Consequently, when PCS is used for print quality evaluation, symbols that offer good reliable performance may fail the minimum PCS requirement and symbols that meet it may not scan reliably.

It is, however, possible to relate PCS measurements to symbol contrast measurements by taking R_L as equal to R_{max} and R_D as equal to R_{min} (an assumption which may not represent the actual measurement of PCS by a given device). PCS and SC may then be calculated from each other as follows.

$$PCS = SC/R_{max}$$

$$SC = PCS \times R_L$$

NOTE Scan reflectance profiles in which PCS < 0,50 will fail the R_{min} test of this document (see 6.2.2) and will therefore be graded 0.

Table H.1 shows the values of symbol contrast and PCS for representative combinations of values of R_{max} and R_{min}. Table H.2 shows PCS values for various symbol contrast values and grades, for a range of values of R_{max}.

Table H.1 — SC and PCS for various reflectance combinations

R_{max} (R_L) / R_{min} (R_D)		90	80	70	60	50	40	30	20
5	SC	85	75	65	55	45	35	25	15
	PCS	0,94	0,94	0,93	0,92	0,90	0,88	0,83	0,75
10	SC	80	70	60	50	40	30	20	10
	PCS	0,89	0,88	0,86	0,83	0,80	0,75	0,67	0,50
15	SC	75	65	55	45	35	25	15	5
	PCS	0,83	0,81	0,79	0,75	0,70	0,63	0,50	0,25
20	SC	70	60	50	40	30	20	10	0
	PCS	0,78	0,75	0,71	0,67	0,60	0,50	0,33	0
25	SC	65	55	45	35	25	15	5	0
	PCS	0,72	0,69	0,64	0,58	0,50	0,38	0,17	0
30	SC	60	50	40	30	20	10	0	0
	PCS	0,67	0,63	0,57	0,50	0,40	0,25	0	0

NOTE Cells to the right of the heavy line represent combinations of values which would be graded 0 in a scan reflectance profile according to this document, because SC < 20, because $R_{min} > (0,5 \times R_{max})$, or both.

Table H.2 — Value of PCS for various values of SC and R_{max} (R_L)

SC (grade) / R_{max}	80 (4)	70 (4)	60 (3)	55 (3)	50 (2)	40 (2)	30 (1)	25 (1)	20 (1)
80	1,0	0,88	0,75	0,69	0,63	0,50	0,38	0,31	0,25
70		1,0	0,86	0,79	0,71	0,57	0,43	0,36	0,29
60			1,0	0,92	0,83	0,67	0,50	0,42	0,33
50					1,0	0,80	0,60	0,50	0,40
40						1,0	0,75	0,63	0,50
30							1,0	0,83	0,67
25								1,0	0,80
20									1,0

NOTE Cells to the right of the heavy line represent combinations of values which would be graded 0 on a scan reflectance profile according to this document, because $R_{min} > (0,5 \times R_{max})$.

H.3 Guidance on grading for applications also specifying PCS

In applications which have based contrast requirements on PCS and need to specify quality grading requirements under this document, the following options may be applied.

a) For symbols with generally high background reflectance, define a minimum overall symbol grade covering all parameters, based on Table H.2, the value of PCS_{min} specified by the application and the range of background reflectances likely to be encountered in the application.

b) For applications where a substantial number of low background reflectance symbols may be used (where R_{max} is, for example, typically less than 45 %), define a minimum grade for all parameters except symbol contrast, and a separate (lower) grade for symbol contrast on a similar basis to a). This minimum grade may need to be marginally higher than would have been provided for under a) in order to offset the effects of the low symbol contrast.

These provisions would be applicable if

— the application specification defines a minimum level of PCS for acceptability, and

— acceptably low levels of scanning problems are encountered with symbols on low-reflectance backgrounds, but complying with the minimum PCS requirements of the specification.

Annex I
(informative)

Process control requirements

I.1 General

This annex describes the application of the scan reflectance profile analysis methodology as a source of feedback useful to the control of the principal variables in the symbol production process. These are, most importantly, element width gain or loss and, secondly, the symbol contrast. The method by which correction is applied is a function of the symbol production method used and is not specified here.

I.2 Process control for repetitive printing

For purposes of process control of symbol production involving

— repetitive printing of the same symbol from the same printing plate or similar material, and

— formal quality assurance procedures designed to ensure consistency of print quality for the image area as a whole throughout the print production run, for example, in the production of printed packaging materials.

The following recommendations may be applied:

— sampling frequency and sample size should be specified as part of the symbol producer's formal quality assurance procedures and should be sufficient to enable detection of significant symbol quality deviations;

— the minimum acceptable symbol grade should be defined;

— the minimum number of scans across each symbol should be determined as set out in I.2, dependent on the variability of the symbol production process and on the amount by which the overall symbol grades achieved exceed the minimum acceptable grade defined in accordance with Clause 6.

In the circumstances provided for by this subclause, equipment designed for online assessment of symbol quality at production speeds may perform the defined number of scans by scanning different positions across a number of symbols produced in sequence within a short time period, and analyse the resulting scan reflectance profiles as though they referred to multiple scans of the same symbol. However, this approach is not an exact substitute for taking samples in accordance with 5.2.4 because it may not sample through the full height of the inspection band.

I.3 Number of scans

The number of scans during initial production runs (with particular combinations of production process or equipment, substrates and other materials) should be as specified in 5.2.5. Once a quality trend has been determined in terms of the excess of the overall symbol grade achieved over the minimum grade for acceptability defined in accordance with Clause 6, the number of scans may be reduced to that shown in Table I.1, in which the columns headed "grade excess" represent the excess of the grade achieved over the minimum acceptable grade.

The number of scans for the first three symbols in any production batch should be based on the expected grade difference as determined by past experience; thereafter, it should be based on the moving average of the grade differences achieved for the latest three symbols measured.

Table I.1 — Number of scans

Minimum acceptable grade					
≥3,5			≥2,5	≥1,5	≥0,5
Grade excess	No. of scans	Grade excess	No. of scans		
≥0,2	2	≥0,4	2	3	3
≥0,1	3	≥0,3	3	4	4
<0,1	5	≥0,15	4	6	6
		<0,15	5	8	10

I.4 Bar width deviation

I.4.1 General

The measurement of average bar width gain or loss has been traditionally used as part of a process control procedure to measure print quality. The average bar width gain or loss should be calculated and expressed either directly in dimensional terms or as a percentage of the X dimension (or, if no X dimension has been specified, of the Z dimension) to provide feedback enabling the printing process to be adjusted, which will lead directly to improved decodability and other grades. This factor is not graded since individual element deviations are taken into account in the decodability assessment.

I.4.2 Two-width symbologies

In the case of two-width symbologies, the achieved wide:narrow ratio N for the symbol is calculated as follows.

$$N = (\text{Average wide bar} + \text{average wide space})/2Z$$

The Z dimension is calculated as follows.

$$Z = (\text{Average narrow bar} + \text{average narrow space})/2$$

Intercharacter gaps should not be included in these calculations.

I.4.3 (n, k) symbologies

In the case of (n, k) symbologies, the achieved Z dimensions are calculated as follows:

$$Z = (\text{average } S)/n$$

where Z, S and n are as defined in 4.2.

I.4.4 Average bar width gain/loss

For either type of symbol, the average bar width gain or loss is then given (as a percentage of X or Z) by

$$G = 100 \times (\sum_b - \sum_i)/(X \times b)$$

where

X (and Z, if necessary) is as defined in 4.2 (see note below);

G is the bar width gain (if G is negative this represents bar width loss);

$\sum b$ is the sum of achieved bar widths;

$\sum i$ is the sum of nominal bar widths (see note below);

b is the number of bars.

NOTE In the above formula, the X dimension is replaced by Z if X is unspecified; nominal bar widths are calculated on the basis of the X (or Z) dimension multiplied by either 1 or N for narrow and wide bars, respectively in a two-width symbology, or the number of modules in the bar in a (n, k) symbology.

I.5 Averaging of grades

As referred to in Clause 6, significant additional information may be gained for process control purposes by taking the means of the individual parameter grades obtained over all scan reflectance profiles for a symbol, in order to ascertain the characteristics of the symbol as a whole while smoothing out the effects of localized variations.

Table I.2 indicates the effect of such an averaging procedure. Figures shown are the grades for the various parameters (except for bar width gain).

Table I.2 — Example of scan reflectance profile grading and parameter averaging for 10 scans of a symbol

Scan no.	1	2	3	4	5	6	7	8	9	10	Parameter mean
Decode	4	4	4	4	4	4	4	4	4	4	4
R_{min}	4	4	4	4	4	4	4	4	4	4	4
SC	4	4	3	3	4	4	3	3	4	3	3,5
EC_{min}	4	4	4	4	4	4	4	4	4	4	4
MOD	2	4	4	3	4	4	2	3	4	4	3,4
Defects	3	2	3	3	4	2	3	3	3	3	2,9
Decodability	3	3	4	3	4	3	3	2	3	3	3,1
SRP grade for scan	2	2	3	3	4	2	2	2	3	3	2,6
Bar width gain (%)	23	10	7	18	15	23	27	18	20	22	18,3

NOTE Since each scan reflectance profile grade is based on the lowest individual parameter grade for that scan, the average of the scan reflectance profile grades is not equivalent to the lowest of the average grades for each parameter shown in the right-hand column of the matrix.

Analysis of the scan reflectance profiles above indicates, first, that there is a significant amount of bar width gain, which should be dealt with by printing adjustments (e.g. reducing head temperature on a thermal printer, or reducing impression pressure on a conventional printing press). This has had some effect on decodability; although all scans have decoded correctly, the symbols might not be suitable for the most critical scanning environment and some re-scans might be required. The defects grade is low, indicating the probable presence of numerous specks and voids or, in the case of a relief printing process, the possibility of "squash" effects on ink distribution across the bar, which would also tend to increase bar widths; these will be visible to the eye.

Examination of individual scan reflectance profile grades indicates some variation, suggesting the possibility of unevenness in the printing of the symbol or that of some irregularity in the element edges.

Bibliography

[1] ISO 2859-1, *Sampling procedures for inspection by attributes — Part 1: Sampling schemes indexed by acceptance quality limit (AQL) for lot-by-lot inspection*

[3] ISO 9000, *Quality management systems — Fundamentals and vocabulary*

[2] ISO/IEC 15426-1, *Information technology — Automatic identification and data capture techniques — Bar code verifier conformance specification — Part 1: Linear symbols*

[4] ISO/IEC 19762, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

ICS 01.080.50; 35.040.50
Price based on 36 pages